VŠB - Technická univerzita Ostrava
Fakulta metalurgie a materiálového inženýrství
Katedra automatizace a počítačové techniky v metalurgii

Zadání bakalářské práce

Student: **Jan Ducháček**

Studijní program: B3922 Ekonomika a řízení průmyslových systémů

Studijní obor: 3902R040 Automatizace a počítačová technika v průmyslu

Téma: Využití integrovaných periferií počítače k diagnostice zařízení

Computer integrated periphery exploitation for device diagnosis

Zásady pro vypracování:

1. Zvukové karty v PC a jejich popis.
2. Možnosti Real-Time vizualizace v programu MATLAB.
3. Možnosti Real-Time zpracování dat v programu LabVIEW.

Seznam doporučené odborné literatury:

1. BEZDĚK, M. Elektronika III. Místo neznámé: Kopp, 2004, str. 236, 80-7232-241-9
2. OPLATKOVÁ, Z., OŠMERA, P., ŠEDA, M., VČELAŘ, F., ZELINKA, I. Evoluční výpočetní techniky - principy a aplikace. Praha: BEN - technická literatura, 2008, str. 536, 80-7300-218-3
3. KAINKA, B., BERNDT, H., J. Využití rozhraní PC pro windows. Místo neznámé: HEL, 2000, 80-86167-13-5

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Milan Heger, CSc.**

Konzultant bakalářské práce: Ing. Robert Frischer

Datum zadání: 30.11.2009

Datum odevzdání: 21.05.2010

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA Fakulta metalurgie a mat. inženýrství 1 OSTRAVA

prof. Ing. Milan Vrožina, CSc.
vedoucí katedry

prof. Ing. Ľudovít Dobrovský, CSc.,Dr.h.c.
děkan fakulty

Zásady pro vypracování bakalářské práce

I.

Bakalářskou prací (dále jen BP) se ověřují vědomosti a dovednosti, které student získal během studia, a jeho schopnosti využívat je při řešení teoretických i praktických problému.

II.

Uspořádání bakalářské práce:

1. Titulní list + zásady pro vypracování BP
2. Prohlášení + místopřísežné prohlášení
3. Abstrakt + klíčová slova česky a anglicky
4. Obsah BP
5. Textová část BP
6. Seznam použité literatury
7. Přílohy

ad 1) Titulním listem je originál zadání BP, který student obdrží na své oborové katedře. Za titulním listem následují tyto „Zásady pro vypracování bakalářské práce“.

ad 2) Prohlášení + místopřísežné prohlášení napsané na zvláštním liste (student jej obdrží na své oborové katedře) a vlastnoručně podepsané studentem s uvedením data odevzdání BP. V případě, že BP vychází ze spolupráce s jinými právnickými a fyzickými osobami a obsahuje citlivé údaje, je na zvláštním liste vloženo prohlášení spolupracující právnické nebo fyzické osoby o souhlasu se zveřejněním BP.

ad 3) Abstrakt a klíčová slova jsou uvedena na zvláštním liste česky a anglicky v rozsahu max. 1 strany pro obě jazykové verze.

ad 4) Obsah BP se uvádí na zvláštním liste. Zahrnuje názvy všech očíslovaných kapitol, podkapitol a statí textové části BP, odkaz na seznam příloh a seznam použité literatury, s uvedením příslušné stránky. Předpokládá se desetinné číslování.

ad 5) Textová část BP obvykle zahrnuje:

– Úvod, obsahující charakteristiku řešeného problému a cíle jeho řešení v souladu se zadáním BP;
– Vlastní rozpracování BP (včetně obrázku, tabulek, výpočtů) s dílčími závěry, vhodně členěné do kapitol a podkapitol podle povahy problému;
– Závěr, obsahující celkové hodnocení výsledku BP z hlediska stanoveného zadání.

BP nemusí obsahovat experimentální (aplikační) část.

BP bude zpracována v rozsahu min. 25 stran (včetně obsahu a seznamu použité literatury). Text musí být napsán vhodným textovým editorem počítače po jedné straně bílého nelesklého papíru formátu A4 při respektování následující doporučené úpravy – písmo Times New Roman (nebo podobné) 12b; řádkování 1,5; okraje – horní, dolní – 2,5 cm, levý – 3 cm, pravý 2 cm. Fotografie, schémata, obrázky, tabulky musí být očíslovány a musí na ně být v textu poukázáno. Budou zařazeny průběžně v textu, pouze je-li to nezbytně nutné, jako přílohy (viz ad 7). Odborná terminologie práce musí odpovídat platným normám. Všechny výpočty musí být přehledně uspořádány tak, aby každý odborník byl schopen přezkoušet jejich správnost. U vzorců, údajů a hodnot převzatých z odborné literatury nebo z praxe musí být uveden jejich pramen - u literatury citován číselným odkazem (v hranatých závorkách) na seznam použité literatury. Nedostatky ve způsobu vyjadřování, nedostatky gramatické, neopravené chyby v textu mohou snížit klasifikaci práce.

ad 6) BP bude obsahovat alespoň 10 literárních odkazu, z toho nejméně 3 v některém ze světových jazyků. Seznam použité literatury se píše na zvláštním listě. Citaci literatury je nutno uvádět důsledně v souladu s ČSN ISO 690. Na práce uvedené v seznamu použité literatury musí být uveden odkaz v textu BP.

ad 7) Přílohy budou obsahovat jen ty části (speciální výpočty, zdrojové texty programu aj.), které nelze vhodně včlenit do vlastní textové části, např. z důvodu ztráty srozumitelnosti.

III.

Bakalářskou práci student odevzdá ve dvou knihařky svázaných vyhotoveních, pokud katedra garantující studijní obor neurčí jiný počet. Vnější desky budou označeny takto:

nahoře: *Vysoká škola bánská - Technická univerzita Ostrava*
Fakulta metalurgie a materiálového inženýrství
Katedra . . . . . .

uprostřed: *BAKALÁŘSKÁ PRÁCE*

dole: *Rok* *Jméno a příjmení*

Kromě těchto dvou knihařky svázaných výtisku odevzdá student kompletní práci také

v elektronické formě do IS EDISON včetně abstraktu a klíčových slov v češtině a angličtině.

IV.

Bakalářská práce, která neodpovídá těmto zásadám, nemůže být přijata k obhajobě. Tyto zásady jsou závazné pro studenty všech studijních programu a forem bakalářského studia fakulty metalurgie a materiálového inženýrství Vysoké školy bánské – Technické univerzity Ostrava od akademického roku 2009/2010.

Místopřísežné prohlášení

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního (§60 – školní dílo);

- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB – TUO) má právo nevýdělečně ke své vnitřní potřebě bakalářskou práci užít (§35 odst. 3);

- souhlasím s tím, že bakalářská práce bude archivována v elektronické formě v databázi Ústřední knihovny VŠB – TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;

- bylo sjednáno, že s VŠB – TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona;

- bylo sjednáno, že užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB – TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB – TUO na vytvoření díla vynaloženy (až do jejich skutečné výše);

- beru na vědomí, že odevzdáním své bakalářské práce souhlasím s jejím zveřejněním podle zákona č. 111/1998Sb., o vysokých školách a o změně a doplnění dalších zákonů (Zákon o vysokých školách) bez ohledu na výsledek její obhajoby.

Místopřísežně prohlašuji, že jsem celou bakalářskou práci vypracoval(a) samostatně.

V Ostravě . . . .

podpis (jméno a příjmení studenta)

Abstrakt

Cílem této závěrečné bakalářské práce je zjistit, zda je možné pomocí osobního počítače nebo případně notebooku provádět některá diagnostická měření. Vše by mělo být realizováno pouze softwarově, odpadá tím drahé hardwarové vybavení pro měření, to znamená, že k osobnímu počítači není připojena žádná externí či integrovaná měřící zásuvná karta a veškerá data jsou získávána přes mikrofonní vstup (Jack 3,5 mm) na zvukové kartě.

Abstract

The objective of this final bachelor thesis is to find out, if it is possible to provide some diagnostic measurement by means of personal computer or notebook. All should be realized only by software, it’s not necessary to use an expensive hardware for measurement, i.e. any external or integrated plug in card is not connected to the personal computer and all data are achieved from microphone analog input (Jack 3.5 mm) on the sound card.

Klíčová slova (Keywords)

Česky (Czech)

MATLAB, LabVIEW, Zvuková karta, Fourierova transformace, rychlá Fourierova transformace, záznam (pořizování) dat, vzorkovací frekvence, ukládání dat, zobrazení dat, reálný čas.

Anglicky (English)

MATLAB, LabVIEW, soundcard, Fourier transform, fast Fourier transform, data acquisition, sample rate, saving data, data view, Real-time

Obsah

1. Úvod

Řešením tohoto tématu je umožnit běžnému uživateli osobního počítače (dále jen PC) či notebooku provádět měření některých veličin bez nutnosti nákupu měřících karet. V této práci bude podrobně popsáno, jak může běžný uživatel pomocí programu MATLAB® firmy The MathWorks ™ v. R2008b 32bit a LabVIEW v. 9.0 64bit firmy National Instruments (dále jen LabVIEW) jednoduše zpracovat data získaná ze zvukové karty ať už zápisem do textového souboru, listu Microsoft® Excel 2003,2007 či vizualizaci získaných dat. Uživatel tak může pomocí svého PC měřit veličiny, které můžeme převést na elektrický signál. Tyto veličiny musí být nejprve převedeny na elektrický signál, který se svede do konektoru Jack 3,5 mm. Poté můžeme pomocí programu LabVIEW data zaznamenat (lze to i pomocí programu MATLAB, ale zpracování je náročnější než v LabVIEW). Pomocí programu MATLAB provedeme vizualizaci signálu. V programu LabVIEW také můžeme vizualizovat data, avšak získané grafy nemůžeme využít v takové šíři jako právě v Matlabu.

2. Teoretický rozbor

Nejprve je nutno definovat základní pojmy, jako jsou A/D (D/A) převodník, vzorkovací frekvence, aliasing a jejich vlastnosti, dále také druhy zvukových karet, vlastnosti daných programů, které byly pro tuto úlohu zvoleny atd.

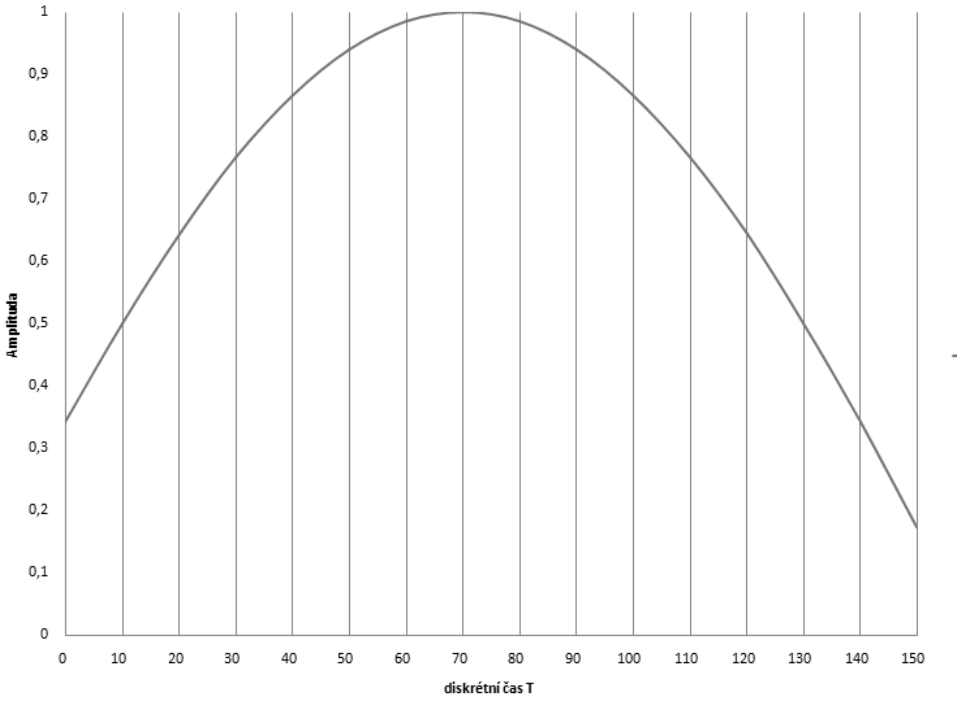
2.1 Vzorkovací frekvence

Vzorkovací frekvence nám slouží k převodu spojitého signálu na signál diskrétní. Tento pojem vychází z Nyquistova vzorkovacího teorému, který říká [6]: *„Rychlost vzorkování musí být přinejmenším dvojnásobkem nejvyšší frekvence, která se vyskytuje ve vzorkovaném spojitém signálu.“* Z tohoto teorému vyplývá, že pokud se zvolí nižší frekvence, nebude signál zrekonstruován přesně. Tento jev nastává v případě, že námi zvolená vzorkovací frekvence je nižší než dvojnásobek maximální frekvence signálu. Dochází tak ke zkreslení rekonstruovaného signálu. Čím bude zvolená vzorkovací frekvence nižší než minimální vzorkovací frekvence, tím bude signál zkreslenější. Princip vzorkování pomocí vzorkovací frekvence je vlastně velice jednoduchý. Mějme nějaký obecný signál o délce 1s. Při vzorkovací frekvenci 22 050 Hz dostaneme z toho signálu 22 050 vzorků, které můžeme dále zpracovávat. Se vzorkovací frekvencí souvisí ještě jeden termín. Je to aliasing. Aliasing

je jev, kdy signály o různém kmitočtu, mohou mít v případě použití určité vzorkovací frekvence stejný průběh. Tato nejednoznačnost lze odstranit zapojením dolnofrekvenčního filtru (antialiasingový filtr).

2.2 A/D D/A převodník

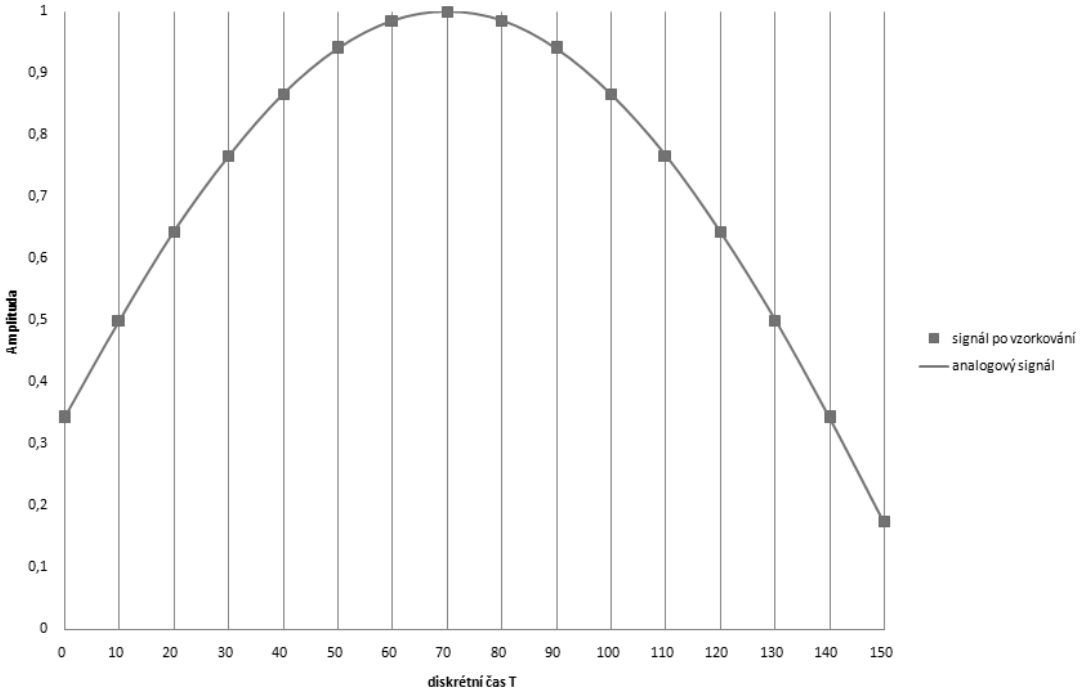
A/D (D/A) převodní ke hardwarové řešení převodu signálu. Písmeno A značí signál Analogový a písmeno D značí signál Diskrétní (neboli také digitální). Toto zařízení nám tedy umožňuje převést spojitý signál na signál diskrétní, v našem případě tok jedniček a nul. Přesnost převodu záleží na mnoha vlastnostech A/D převodníku. Jedním z prvních je už výše zmíněná vzorkovací frekvence. Čím je vyšší, tím přesnější diskrétní signál dostaneme, dále přesnost omezuje převodník díky kvantování, kdy na ose y musí být celočíselné hodnoty a chyba je poté ±0,5 intervalu kvantování. V našem případě je signál převáděn do dvojkové soustavy a velikost kvantování určuje počet bitů převodníku. Jednoduchý princip převodníku si ukážeme na malém příkladu. Mějme nějaký zvolený analogový signál (Obr. 1). Dále si pro ukázku zvolíme, že budeme mít 3 bitový převodník.



Obr. 1 Zvolený analogový signál

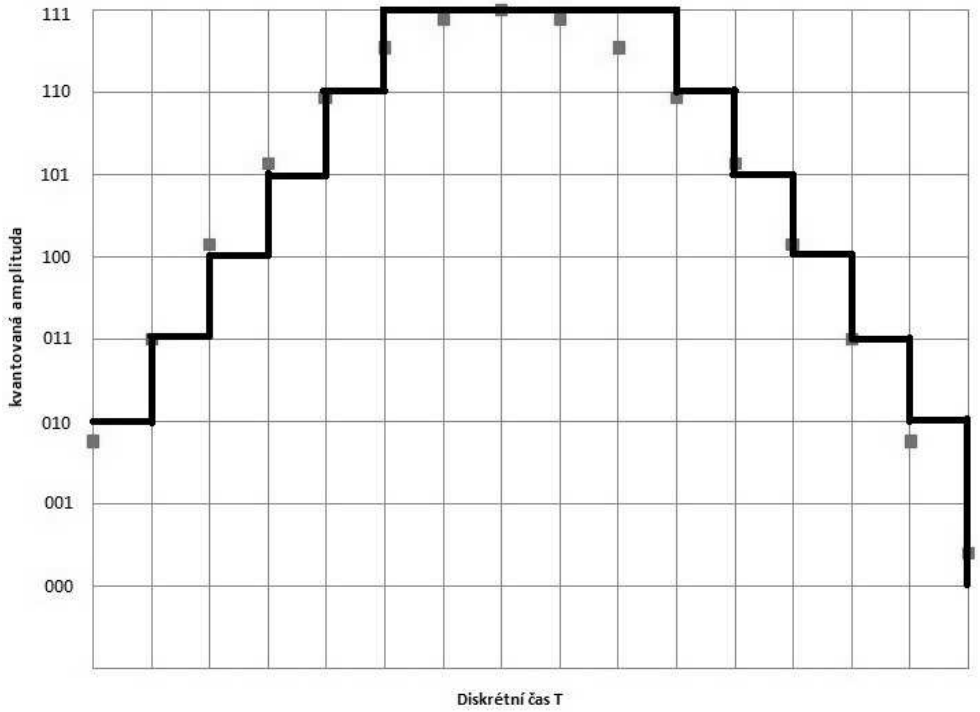
Nejprve v převodníku proběhne vzorkování, tzn. odečteme hodnotu amplitudy signálu v diskrétním čase o počtu, který odpovídá převrácené hodnotě vzorkovací frekvence. Vzorek

je tedy hodnota amplitudy v diskrétním čase T. Graf, jak může vypadat takto zpracovaný signál zobrazuje Obr.2.



Obr. 2 Vzorkování v diskrétním čase T

Poté provedeme kvantování osy y (amplitudy). Uděláme to tak, že osu y rozdělíme na intervaly a jednotlivým intervalům přiřadíme bitové hodnoty. Poté určíme, ke které hodnotě má vzorek blíže a signál nabude hodnoty dané kvantováním. Viz. Obr. 3.



Obr. 3 Signál po vzorkování a kvantování

Výstupní signál nyní vypadá jako skokový. Na výstupu z A/D převodníku se nám poté ukáže pouze sled binárních čísel, nijak neoddělených. Výsledek je poté zpracován, zpravidla počítačem. Dále je několik druhů A/D převodníku [14]. Prvním z nich je převodník paralelní. Je nejrychlejší, ale z důvodu přesnosti je toto řešení nákladné. Vyrábějí se s přesností 6-10 bit a vzorkovací frekvence se pohybují mezi 10 MHz a 3 GHz. Dalším typem je převodník s postupnou aproximací. Nevýhodou u tohoto typu je rychlost. Čím větší bude přesnost, tím bude rychlost převodu nižší. Vyrábějí se 8-16 bit a pro vzorkovací frekvence 30 kHz až 3 MHz. Dalším typem je převodník s dvojitou integrací. Je pomalý, ale velmi přesný a odolný proti šumu. Vyráběn 10-27 bit a pro frekvence 0,1 Hz až 1 kHz. Dále existují převodníky sigma-delta. Vyráběny v přesnosti 16-24 bit a pro frekvence 10 Hz až 100 kHz. D/A převodník pracuje na principu opačném než A/D a uplatňuje se především v osobních přehrávačích, PC apod. Výsledný analogový signál je ovšem vždy zkreslený díky diskrétním skokům, které jsou způsobeny kvantováním a vzorkováním.

2.3 Zvukové karty

Zvukové karty jsou v dnešní době nedílnou součástí PC. Lze je rozdělit [8] např. podle typu sběrnice: Určené pro sběrnice ISA, dnes již téměř nepoužívané a na ty, které jsou určeny pro sběrnici PCI. Dále je lze rozdělit podle konstrukce na integrované (jsou přímo součástí základové desky) a rozšiřující karty, které jsou připojeny přes sběrnici PCI. Zvuková karta pracuje v akustickém spektru od 20 Hz do 20KHz tedy v pásmu zvuku zachytitelného zdravým lidským uchem. Většina zvukových karet využívá pro komunikaci kanál DMA, který pracuje v režimu full duplex, neboli je schopen nahrávat a přehrávat signál najednou. Podle druhu karty může být zvuková karta tvořena jedním nebo dvěma systémy. Všechny tyto systémy jsou tvořeny párem převodníků, A/D převodníkem a D/A převodníkem. A/D převodník slouží pro záznam zvuku a D/A pro přehrávání. Jeden z těchto systémů je určen čistě pro záznam a reprodukci zatímco druhý se využívám pro zvuk MIDI (samplovaný zvuk) a je určen k hudební syntéze. Druhý se vyskytuje pouze u desek s MIDI konektorem (v dnešní době tento konektor má většina zvukových karet). Označení konektorů [7] nám ukazuje Tab. 1

BARVA KONEKTORU	NÁZEV KONEKTORU	ÚČEL
Modrá	Line-in	Slouží ke vstupu z externích zařízení
Růžová	Mic-in	Slouží jako vstup pro mikrofon
Zelená	Line-out	Výstup pro připojení aktivních reproduktorů nebo výstup pro připojení k zesilovači.
Šedá	Speaker-out	Používá se pro připojení sluchátek, pasivních reproduktorů, je zde k dispozici zesílený zvuk o výkonu 4-8 W
Černá	Joystick	Umožňuje připojení Joysticku a označuje se jako MIDI (game port)
Šedá	Audiokonektor (AUX)	Připojuje se do něj kabel z CD mechaniky

Tab.1 Tabulka označení vstupů a výstupů zvukové karty

2.4 Stručná charakteristika MATLABu

Program MATLAB® je produktem firmy The MathWorks ™ (dále jen MATLAB). V tomto programu budeme provádět Real-Time vizualizaci dat získaných ze zvukové karty. Po prozkoumání literatury [1] [2] jsem došel ke dvěma řešením. Prvním z nich je nahrát data z analogového vstupu zvukové karty vytvořením analogového vstupu příkazem *analoginput ('winsound').* Poté je můžeme přímo vizualizovat nebo je zpracovat a poté vizualizovat. Druhým řešením, mnohem elegantnějším a rychlejším, se mi zdá příkaz *softscope*. Důležitým upozorněním je, že tento příkaz se nachází pouze ve 32 bitové verzi tohoto programu a v 64 bitové verzi jej spustit nelze. Tímto příkazem se v matlabu otevře již naprogramovaný jednoduchý osciloskop. V úvodním okně, které se nám otevře, navolíme hodnoty pro náš vstup, jako např. ID vstupu, vzorkovací frekvenci a počet kanálů. Poté bude na nás, abychom si určili délku měřeného signálu, avšak je nutno brát v úvahu kapacitu této použité funkce. Jako příklad lze uvést toto. Vzorkovací frekvenci nastavíme na 44 100 Hz, kdybychom chtěli zaznamenat 3 s (132 300 vzorků), tak nebudeme mít kompletní záznam a to z důvodu

možnosti záznamu pouhých 100 000 vzorků. Takže pro tuto frekvenci je vhodné použít délku záznamu 2 s neboli 88 200 vzorků. Můžeme uložit získaná data do proměnné a můžeme dále zpracovávat. (Zobrazovat získané hodnoty v grafu, použít Fourierovy transformace, apod.)

2.5 Stručná charakteristika LabVIEW

Program LabVIEW je navržen tak, aby si uživatel mohl snadno vytvořit svůj virtuální přístroj a pomocí něj zpracovávat data získaná z externích zařízení. Tento program byl vyvinut především pro záznam a zpracování dat s tzv. DAQ zařízení (Data Acquisition = pořizování dat) neboli externích či rozšiřujících zásuvných měřících karet. Avšak i tento program umožňuje zpracovávat data získána ze zvukové karty. V programu LabVIEW jsou dvě vývojová okna. První je Čelní panel (neboli Front Panel), který nám vlastně ukazuje, jak by námi vytvořený přístroj mohl vypadat ve skutečnosti. Také se na něm nalézají hlavní ovládací prvky. Druhé okno se nazývá Blokový diagram (Block Diagram), ten slouží zaprvé jako schéma našeho přístroje a za druhé jako vnitřní část přístroje, nalezneme zde operační jednotky, vedení, napojení na zobrazovače či ovladače. Pro náš případ bylo nejtěžší provést zápis dat získaných ze zvukové karty do souboru, protože získaná data byla jiného typu, než vyžadovala funkce uzpůsobená k zápisu dat do souboru. Tento problém se vyřešil tak, že pole dat ze zvukové karty je převedeno na *cluster* (neboli datová struktura, která, na rozdíl od pole, může obsahovat více druhů datových typů) a tento *cluster* poté pomocí funkce *unbundle* signál rozloží tak, že jej již můžeme zpracovat. Poté pomocí funkce *Write To Spreadsheet File* můžeme uložit námi naměřené hodnoty do souboru, který potřebujeme (např. *.xls či *.m)

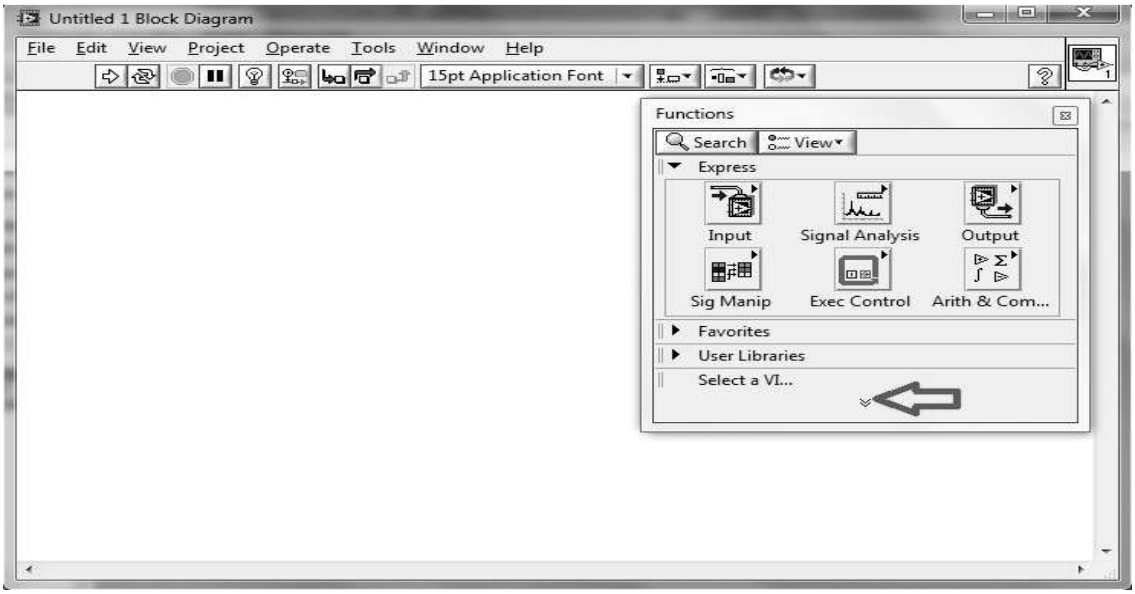
3. Návrh nastavení softwaru k záznamu dat ze zvukové karty

V této práci budu popisovat realizaci nastavení komponent a příkazů v obou programech a poté na těchto programech provedeme kontrolní měření vibrací. Výsledky měření vibrací budou porovnány s výsledky naměřenými pomocí specializované měřicí karty. Toto porovnání nám ukáže, jak přesná je naše metoda. Veškerá měření probíhala na následující sestavě. Hardware: Dvoujádrový procesor T4300 (2,1 GHz), 3GB DDR3 RAM, grafická karta Intel Graphic Media Accelerator 4500M a zvuková karta Conxant High

Definition SmartAudio 221. Software: Windows 7 64bit, MATLAB v. 7.7.0.471 32bit, LabVIEW 9.0 64bit.

3.1 Návrh virtuálního přístroje v LabVIEW

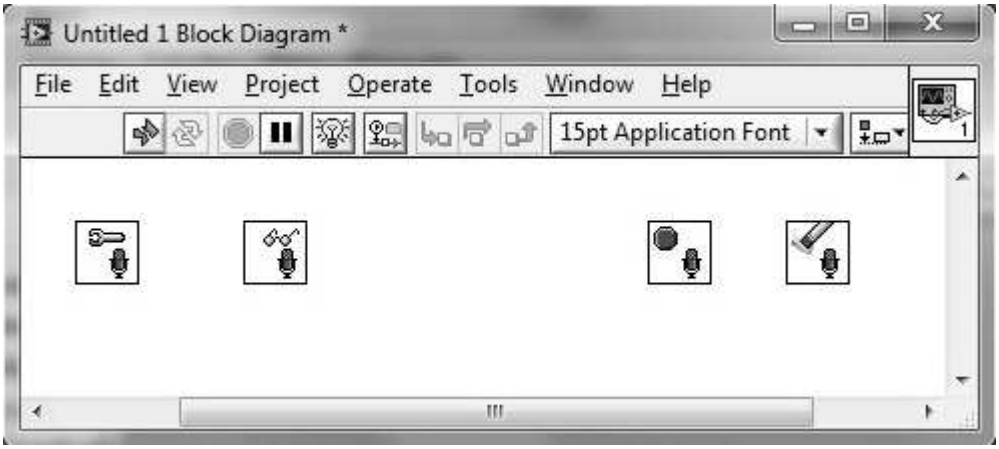
Pro začátečníky v tomto programu je výhodné zvolit vhodnou literaturu [4] pro zvládnutí základního ovládání programu. Tento program slouží k vytvoření virtuálních měřících přístrojů, které nám, při patřičně upravených vstupech, mohou sloužit jako opravdové přístroje. Po spuštění programu se nám otevře obrazovka, ve které zvolíme možnost Blank VI. Poté se nám otevřou dvě, respektive tři okna. Poté pomocí kombinace kláves CTRL+E přepneme okna na blokový diagram (*Block diagram*) a zobrazíme si funkce programu. Použijeme View > Functions Palette. Abychom měli přehled o všech dostupných funkcích, použijeme tlačítko ve spodní části okna *Functions*, jak nám ukazuje obr. 4.



Obr. 4 Ukázka rozevření palety *Functions*

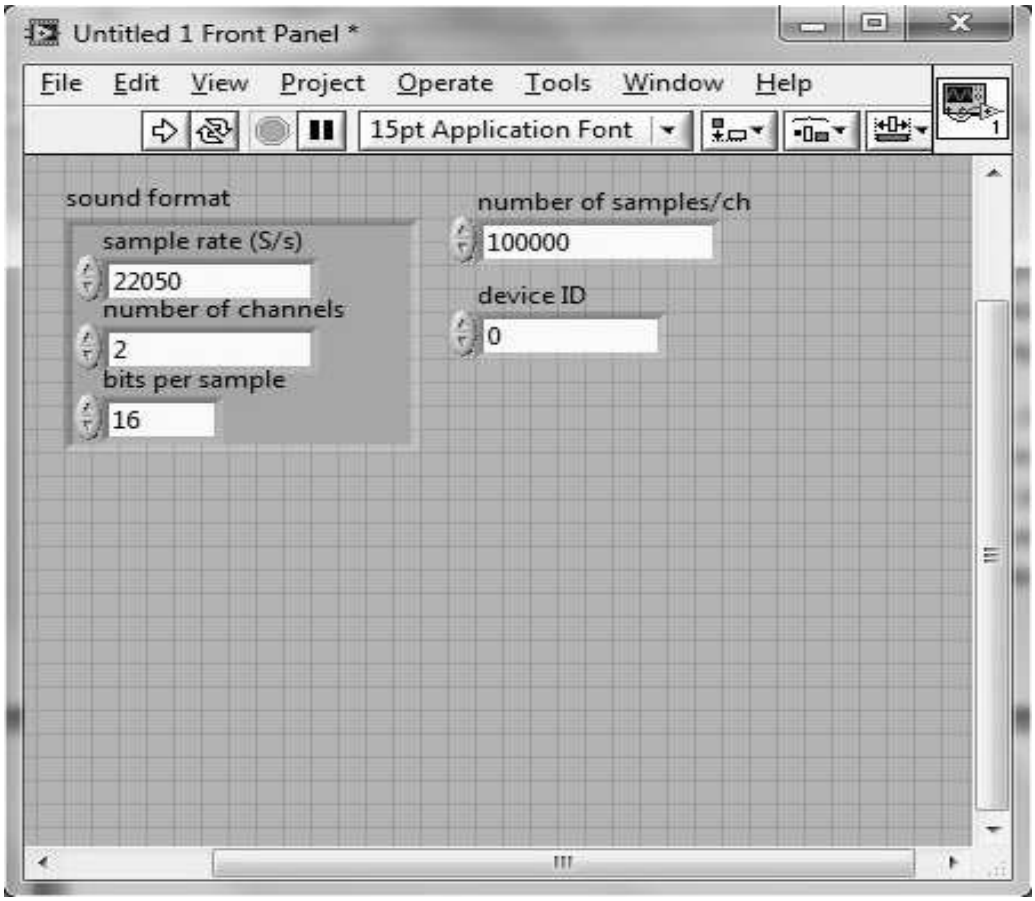
Nyní potřebujeme přidat do programu jako zdroj vstupních dat zvukovou kartu. Toto propojení realizujeme pomocí několika funkcí. Musíme začít vložením funkce *Configure*, kterou nalezneme pod Programming > Graphics&Sound > Sound > Input. Tato funkce nám umožní definovat parametry zaznamenávaného signálu. Nyní potřebujeme funkci, pomocí níž načteme vzorky ze zdroje. K tomuto slouží funkce *Read*, kterou nalezneme ve stejné kategorii jako *Configure*. Tato funkce nám umožňuje zadat délku záznamu a počet vzorků. Dále se vloží funkce *Stop*, která, jak už napovídá název, ukončuje načítání vzorků ze zdroje. Nakonec budeme potřebovat funkci *Clear*, která uvolňuje paměť, abychom dále zbytečně nezatěžovali systém. Nyní máme jednotlivé funkce umístěny v blokovém diagramu v jedné řadě, jak nám

ukazuje obr. 5. Teď musíme jednotlivé komponenty spojit. Provedeme to pomocí komponenty *Connect Wire*  , kterou nalezneme na paletě Tools. Spojení bude následující.

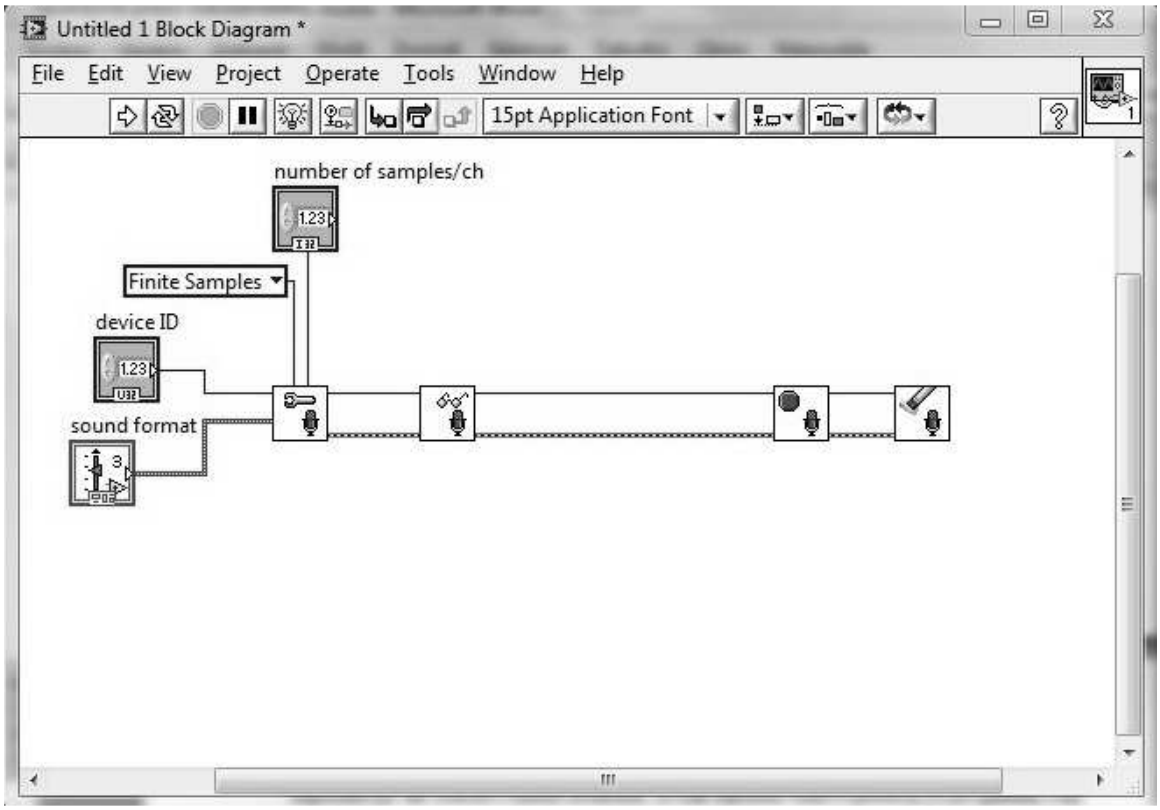


Obr.5 Umístění komponent *Graphic & Sound*

Výstup *Configure Task ID* bude vstupem *Task ID Read*. Výstup *Task ID Read* bude vstupem *Task ID Stop* a výstup *Task ID Stop* bude vstupem pro *Task ID Clear*. Ve stejném pořadí spojíme chybové vedení tzv. *error in (no error)* a *error out*. Nyní musíme konfigurovat vstupní signál a to provedeme tak, že pořád máme aktivní tlačítko *Connect Wire*. Pravým tlačítkem myši klikneme na vstup *Configure Sound format* (Zvukový formát) a v kontextovém menu zvolíme Create > Control. Dále klikne pravým na *Device ID* (ID zařízení) a opět v kontextovém menu vybereme možnost Create > Control. Tutéž akci provedeme pro *Number of samples/ch* (Počet vzorků). Poté klikneme na *Sample mode* (Vzorkovací mód) a vybereme možnost Create > Constant. Pro vzorkovací mód jsme nastavili vstup jako konstantu a defaultně je nastaven na *Continuous Sample* (pokračující vzorky), avšak pro naše řešení je vhodnější nastavit konečný počet vzorků přenastavením na *Finite Samples*. Abychom mohli měnit hodnoty v této konstantě, musíme přepnout na *Operate Value*  . Když nyní přepneme pomocí Ctrl+E na *Front panel*, zjistíme, že nám tam přibyly 3 nové komponenty. Tyto komponenty nám reprezentují vytvořené (Create) ovladače (Control). Ty si můžeme libovolně posouvat a uspořádávat, aniž bychom měnili jejich pozici v *Block diagramu*. Takže nyní máme podstatě hotov virtuální přístroj, který nám po spuštění zaznamená signál ze zvukové karty. Ovšem tento přístroj nám zaznamenaná data nikde neprezentuje ani dokonce nikam neukládá. Avšak základní funkce přístroje je již splněna. Náš přístroj by měl nyní vypadat jako na Obr. 6 a Obr. 7.



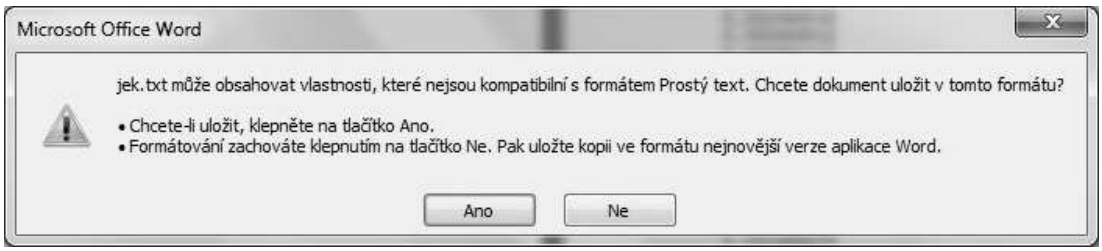
Obr. 6 Čelní panel



Obr. 7 Blokové schéma

Důležitým prvkem záznamu signálu jsou jeho parametry. Jak už jsme si řekli v úvodu jedna z nejdůležitějších vlastností přístroje je vzorkovací frekvence. Tu nastavíme v čelním panelu v položce *Sample rate (S/s)*. Nastavujeme ji zde v jednotkách Hertz (Hz). Dalším parametrem, který musíme nastavit je počet kanálů (*number of channels*). A v neposlední řadě musíme nastavit, s jakou přesností bude vzorek zaznamenáván. Toto provedeme pomocí specifikace počet bitů na vzorek (*bits per sample*). Nastavení veškerých parametrů závisí na parametrech, které je schopna zpracovat zvuková karta. V našem případě si můžeme dovolit natavit vzorkovací frekvenci na 96 000 Hz, 24 bitů na vzorek a pro účel našeho měření postačí 1 kanál. Nyní musíme vyřešit problém prezentace respektive uložení pořízeného záznamu. Výstup zaznamenaného signálu nalezneme u komponenty *Read* s názvem *Data*. Pro ukládání dat nám slouží komponenta *Write To Spreadsheet File*, kterou nalezneme pod *Programming > File I/O*. Tato komponenta nám umožňuje zaznamenaná data ukládat i například ve formátu *.xls či *.xlsx což jsou formáty aplikace MS Excel. Po umístění *Write To Spreadsheet File* do blokového diagramu napojíme výstup *Read Data* na vstup *1D data Write To Spreadsheet File*. Když nyní spustíme program, ukáže se nám možnost uložení souboru. Vybereme umístění, kde chceme námi získaná data uložit, poté si zvolíme název a při uvádění názvu nesmíme zapomenout na koncovku souboru. Pokud napíšeme pouze název bez koncovky, tak bude soubor uložen jako soubor obecného typu a počítač nebude vědět v jakém programu jej musí otevřít. Jako koncovka se nám nabízí několik typů. Zaprvé je to koncovka *.xls určená MS Excel 2003. Jeví se to jako nejjednodušší možnost, avšak je tu důležité omezení použití tohoto typu a to je, že aplikace MS Excel 2003 zvládne zapsat pouze 65 536 vzorků. Další možností by bylo uložit data pod koncovkou *.xlsx, která patří pod MS Excel 2007. Zde je jejich počet omezen až na 1 048 576. Tato koncovka stejně jako *.xls je vhodná pouze pokud nechceme dále data zpracovávat (zobrazovat v grafech, upravovat), protože např. pokud bychom chtěli vytvořit 2D graf jsme limitování hodnotou 32 000 bodů. Pokud je bodů více musí se vytvořit několik grafů. Dalším typem je koncovka *.mat, která by nám umožňovala data zpracovávat pomocí MATLABu. Při této koncovce narazíme na úplně jiný problém a to, že většina dat má desetinnou hodnotu a tak se ukládá ve formě s desetinnou čárkou. Bohužel MATLAB využívá desetinnou tečku, a proto zaznamenaná data nepřečte správně. Nyní koncovka *.txt. Tato koncovka patří prosté - mu textu a bude pro nás nejvýhodnější hned z několika důvodů. Koncovku *.txt můžeme načíst, jak do aplikace MS Excel, tak do MATLABu a také do mnoha dalších programů. Než začneme ukládat, musíme nadefinovat, v jakém formátu budou naměřené vzorky ukládány. K tomu nám slouží vstup *Write To Spreadsheet File format (%.3f)*. Abychom mohli formát definovat, musíme tento vstup

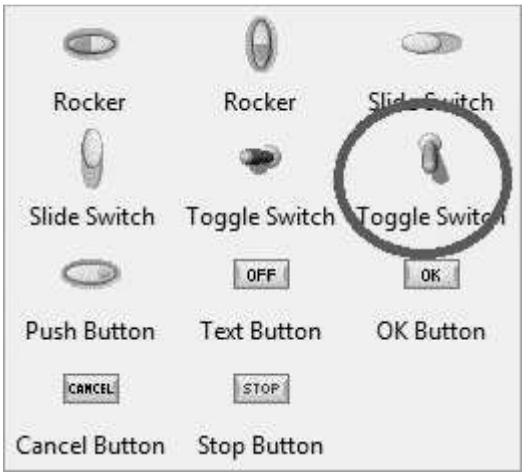
nastavit jako *Constant* a to již známým postupem pomocí *Create*. Výraz obsažený dříve v závorce se nám objeví jako konstanta. Tento výraz využívá syntaxe jazyku C a říká nám, že proměnná (vzorek) bude typu %f, což znamená *Float* neboli reálné číslo s plovoucí desetinnou čárkou a bude zaokrouhlena *.3* neboli na 3 desetinná místa. Abychom mohli lépe s naměřenými daty pracovat, bude nutné tento formát změnit. Pro naše měření je vhodné nastavit proměnnou na reálné číslo s plovoucí desetinnou čárkou, zaokrouhlené na devět desetinných míst a každý další vzorek se bude zapisovat na nový řádek (bude se tvořit sloupec). Tento formát docílíme pomocí této syntaxe: *%.9f\n*. Znak na konci výrazu (*\n*) programu říká *new line* neboli nový řádek. Nyní po spuštění programu dojde k záznamu dat. Po nashromáždění všech požadovaných vzorků se nám objeví opět okno, kde nastavíme místo uložení a typ souboru (*.txt). V této fázi musíme vhodně nahradit desetinné čárky za tečky. Jedna možnost, která je však extrémně pracná, je provést to ručně. Tato možnost je vhodná pro velmi malý počet vzorků. Pokud, jako v našem případě, je vzorků opravdu mnoho můžeme si pomoci aplikací MS Word. Aplikaci MS Word si spustíme a zvolíme možnost otevřít soubor. Vybereme z námi zvoleného umístění soubor, který jsme vytvořili pomocí LabVIEW. Počkáme, až jej MS Word načte a poté pomocí funkce nahradit zaměníme desetinné čárky za tečky. Provedeme to tak, že do pole Najít napíšeme "," a do pole Nahradit čím napíšeme ".". Klikneme na možnost Nahradit vše, chvíli počkáme a objeví se nám okno, které nás informuje, kolik nahrazení proběhlo. V našem případě by se tam mělo objevit číslo 96 000 (reprezentující námi zvolený počet vzorků). Poté soubor uložíme. Aplikace MS Word nám zobrazí varování o změně formátu, jako nám ukazuje Obr. 8.



Obr.8 Potvrzení formátu textu

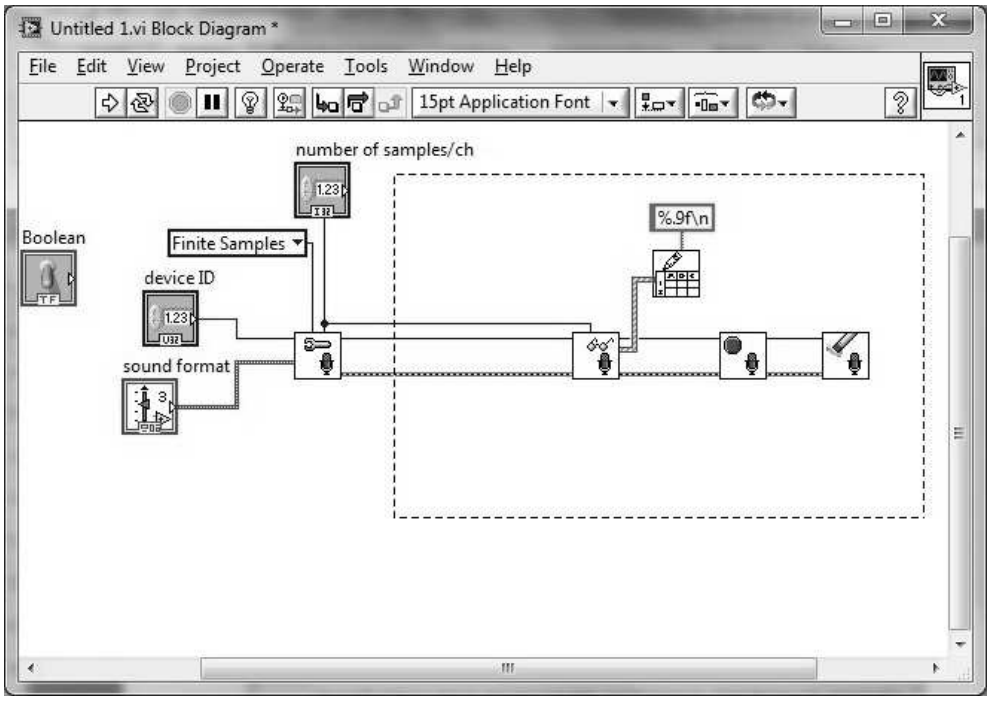
Zvolíme možnost Ano. Nyní máme soubor s daty, která obsahují desetinnou tečku. Tyto data si můžeme načíst do MATLABu a dále zpracovávat. Postup si ukážeme později, když se budeme věnovat MATLABu. Dalo by se říci, že námi vytvořený přístroj v LabVIEW by pro někoho mohl být hotov. Prakticky tomu tak je, ale je vhodné nastavit ještě několik úprav. Například záznam proběhne hned po spuštění a někdy to není vhodné. Budeme se nyní věnovat tomu jak program spustit pomocí čelního panelu. K tomu použijeme komponentu

Toggle Switch. Tuto komponentu musíme umístit na čelní panel z palety *Controls*. A najdeme ji *Express* > *Buttons & Switches*. Jak zjistíme, jsou tam dvě tyto komponenty a v našem případě použijeme tu, kterou znázorňuje Obr. 9



Obr. 9 Umístění komponenty *Toggle Switch*

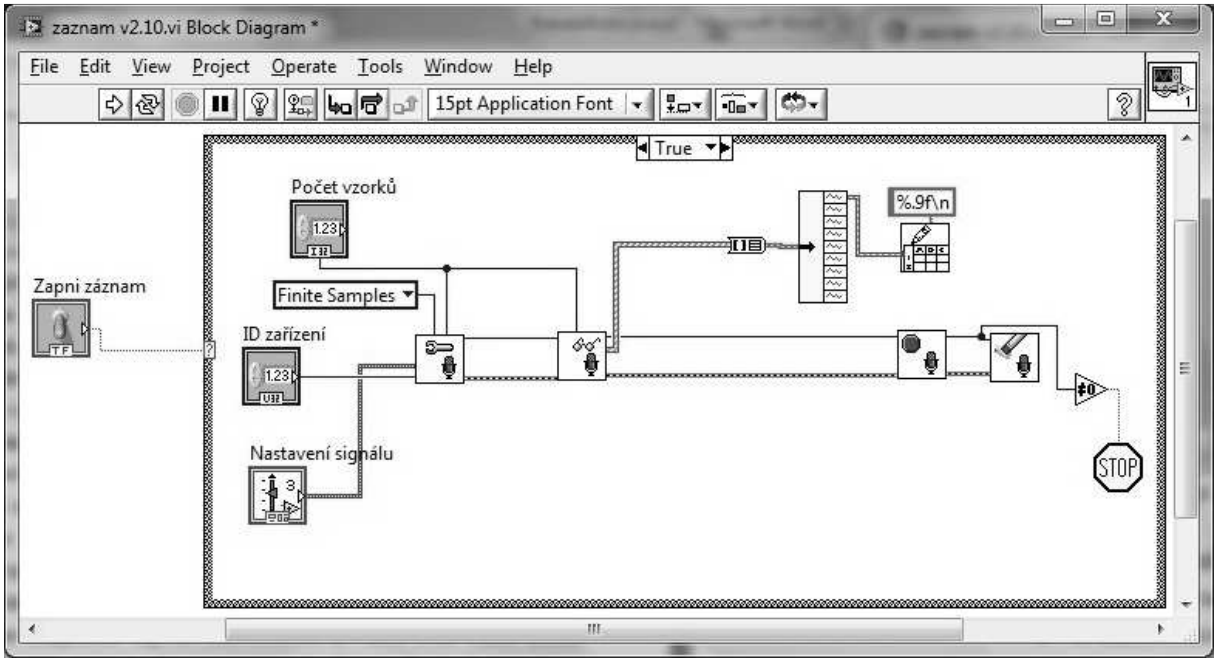
Samotný tento *Button* (tlačítko, knoflík) nám ovšem v programu nic neprovede a musí být provázán s nějakou strukturou. Jeho vlastností je, že na výstupu indikuje hodnotu *True* a *False*, neboli Pravda a nepravda, či 1 a 0. Když nahlédneme do literatury [11] zjistíme, že vhodnou strukturou pro náš problém bude struktura *Case Structure*, jak se v publikaci dočteme tato struktura má stejné vlastnosti jako v mnoha dalších jazycích jako je C, C++, Pascal, potažmo Delphi apod. Strukturu nalezneme v paletě *Functions*. K této struktuře se můžeme dostat dvěma způsoby. Prvním je najít ji v *Express* > *Execution Control*. Je také umístěna v *Programming* > *Structures*. Abychom do této struktury zahrnuli námi vytvořené prvky musíme při vkládání struktury kliknout do blokového diagramu a pomocí čárkovaného obdélníku vybrat potřebné komponenty (viz. Obr. 10). V struktuře musí být zahrnuty tyto všechny komponenty. Vstupem do této struktury bude použit náš „knoflík“ (*Toggle Switch*), který napojíme na malý zelený otazníček ve středu levé strany struktury. Důležité je, že *Case Structure* má jako předdefinovaný typ dvě vrstvy *True* a *False*. Pro přepínání mezi vrstvami použijeme komponentu *Operate Value*. Pokud bychom na vstup *Case Structure* (na zelený otazníček) připojili například číslo typu *Integer* (celé číslo) vrstvy by se nám změnily z hodnot *True* a *False* na celočíselné hodnoty + hodnota *Default*. Hodnota *Default* nám určuje, co se bude dít, pokud na vstup přivedeme jinou hodnotu než ošetřenou v předchozích vrstvách. Tato varianta ovšem pro nás nemá velký význam, protože *Toggle Switch* nabývá pouze hodnot *True* a *False*, což je předdefinovaný typ struktury. Nyní tedy máme zapojenou strukturu *Case* a s ní i komponentu *Toggle Switch*. Pokud je *Toggle Switch* v poloze *False*, tak



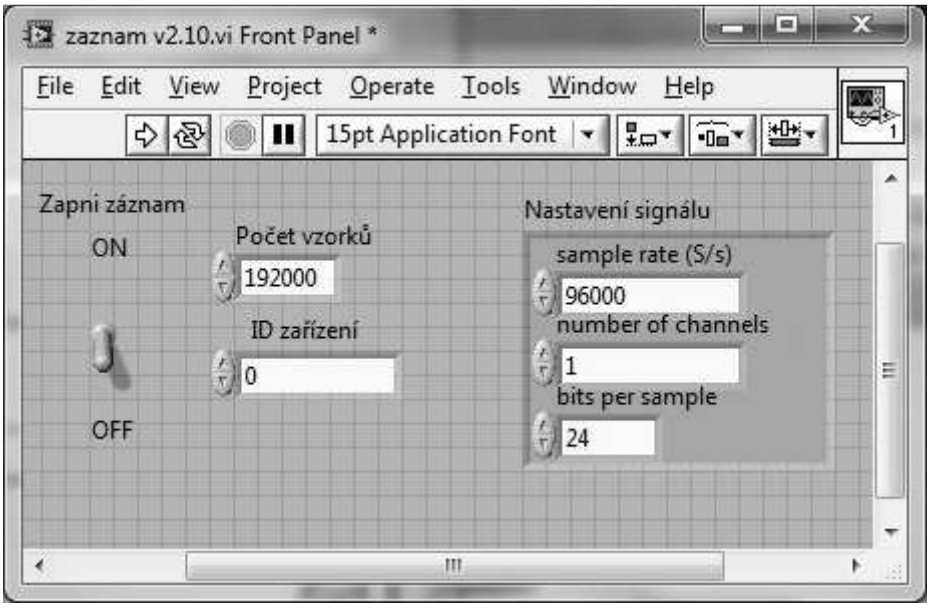
Obr. 10 Vybírání *Case Structure*

po spuštění programu se nestane nic. Nedostaneme ani možnost přepnout při běhu programu *Toggle Switch* do polohy *True*. Musíme proto při spouštění zvolit možnost *Run Continuously* („běh dokola“). Nyní se nám bude pořád opakovat program, dokud nepřepneme „knoflík“ do polohy *True*. Poté se nám tedy spustí námi vytvořený program. Dojde k záznamu a vyskočí nám obrazovka, kde budeme nastavovat umístění a typ souboru. Po uložení nám za chvíli vyskočí tato obrazovka znova. Toto se děje právě díky spuštění pomocí *Run Continuously*. Abychom tomu předešli, protože nám stačí měřit na jeden průchod, musíme do programu vložit komponentu tak, aby po zaznamenání dat byl běh programu ukončen. K tomu slouží komponenta *Stop*, kterou nalezneme v *Programming* < *Application Control*. Vstup této komponenty je nastaven na vyhodnocení hodnot *True* a *False*, proto musíme z programu tuto hodnotu nějak dostat. Nejprve umístíme komponentu *Stop* do *Case Structure* do vrstvy *True*. Pokud bychom napojili vstup do této struktury přímo na komponentu *Stop*, došlo by ihned k vypnutí programu a nedošlo by k zaznamenání žádných dat. Z tohoto důvodu musíme do *Case Structure* vložit ještě jednu komponentu. Tato komponenta se nazývá *Not Equal To 0?*. Tato komponenta zjišťuje, zda přivedený vstup je nenulový, pokud ano vrací hodnotu *True*, pokud je přivedená hodnota nulová, vrací hodnotu *False*. Na vstup této komponenty napojíme výstup *Task ID* komponenty *Stop*, která nám slouží k ukončení pořizování záznamu dat ze zvukové karty. Tato hodnota bude vždy po průchodu nenulová a tak nám umožní až po záznamu program ukončit. Vrstva *False* bude ponechána prázdná, protože když je tlačítko

vypnuto, chceme, aby přístroj nic nedělal. Na Obr. 11 je tedy zobrazeno, jak bude vypadat blokové schéma námi vytvořeného virtuálního přístroje. Na Obr. 12 je naopak zobrazen čelní panel našeho přístroje, kde můžeme nastavovat vstupní hodnoty zaznamenávaného signálu. Čelní panel slouží také k ilustraci, jak by mohl náš přístroj vypadat v reálném provedení.



Obr. 11 Blokové schéma

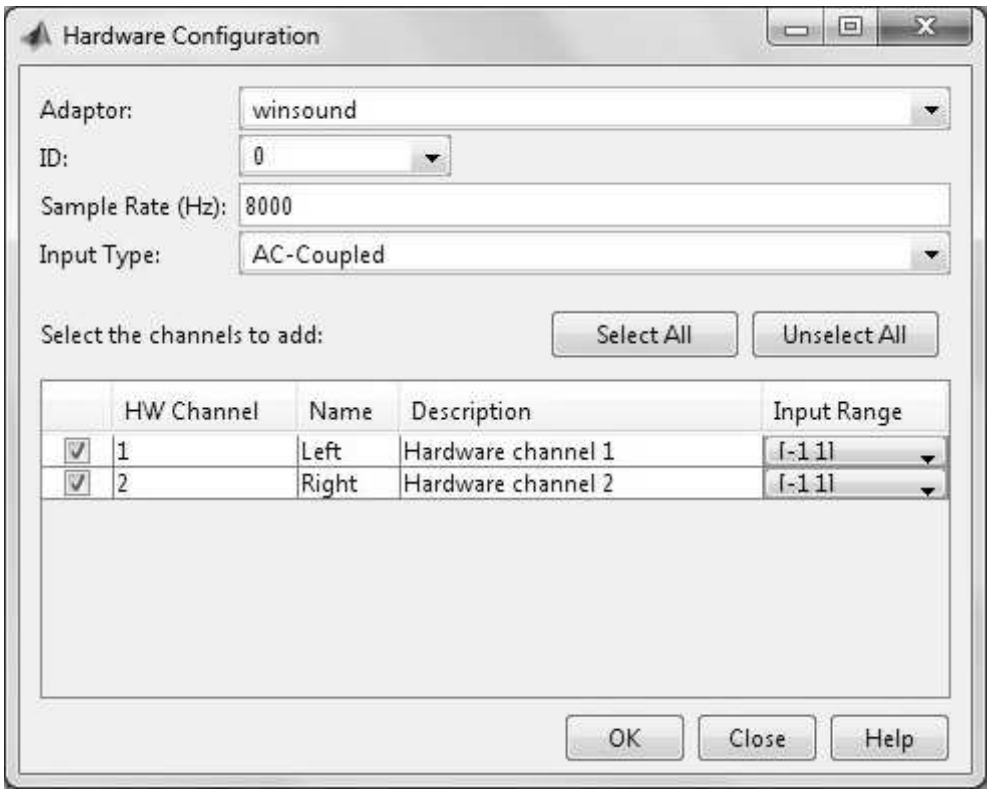


Obr. 12 Čelní panel

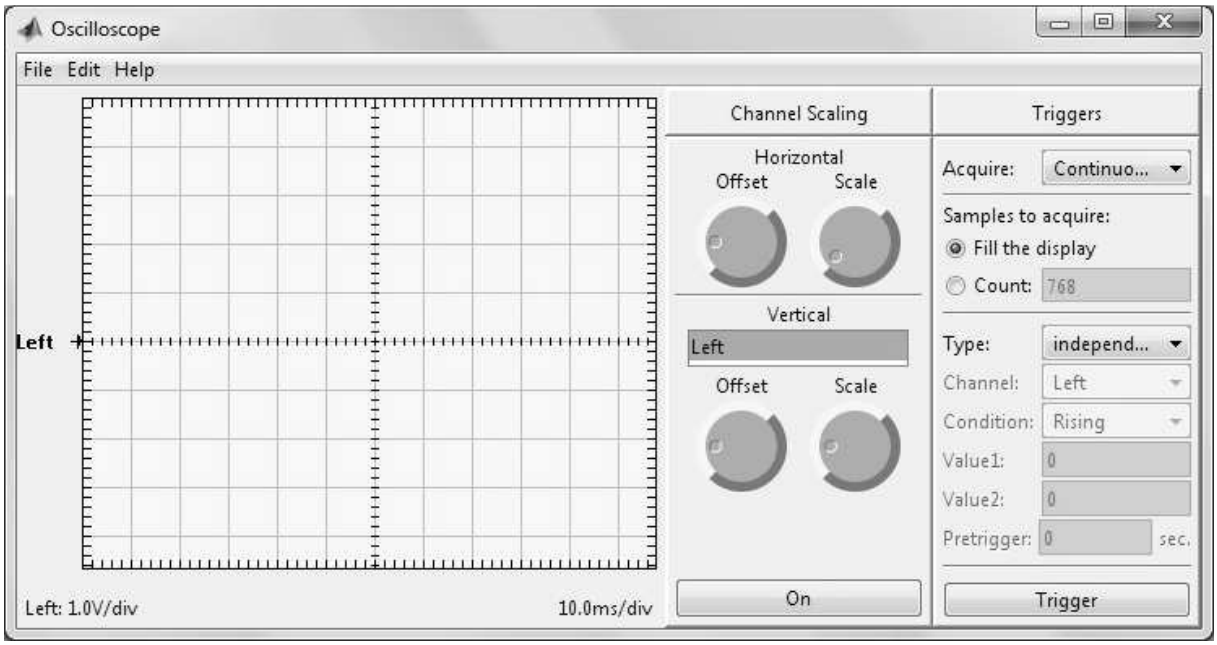
3.2 Možnosti záznamu dat ze zvukové karty v MATLABu

Tento program má velmi široké využití v různých oblastech, od zpracování signálů, přes počítání matic, výpočtu integrálu a mnoha dalších jednodušších operací a končící prací s neuronovými sítěmi. Úplným začátečníkům doporučuji před dalším čtením této práce nastudovat základní literaturu [5], kde se naučí základní operace v MATLABu a také je vhodné nastudovat literaturu ohledně zpracovávání signálů v MATLABu [2]. V tomto programu existují dvě cesty [1], kterými se je možno vydat. Každá z nich má své výhody a nevýhody, které zde budou porovnány. První a jednodušší cestou je použití příkazu *Softscope*. Tento příkaz v MATLABu spustí jednoduchý osciloskop, pomocí kterého provedeme záznam dat ze zvukové karty. Po zadání tohoto příkazu se nám zobrazí okno (Obr. 13), kde provedeme konfiguraci záznamu. V tomto okně ponecháme nastavenou hodnotu *Adaptor* na *winsound*, *ID* na *0*, *Sample Rate (Hz)* nastavíme na hodnotu 96 000, *Input Type* ponecháme na hodnotě *AC-Coupled* a v tabulce necháme zatržen pouze *HW Channel 1*. Tímto jsme provedli nastavení následujících parametrů. Signál se bude načítat z adaptéru zvukové karty, načítat se bude z mikrofonového vstupu, vzorkovací frekvence bude 96 000 Hz, vstupní typ bude střídavá vazba (kanál zvukové karty), a zatržením pouze *HW Channel 1* budeme snímat signál pouze z jednoho kanálu. V našem případě levého (*left*). Poté klikneme na *Ok*. Konfigurační okno bude nahrazeno oknem s virtuálním osciloskopem (Obr. 14). Pokud budeme chtít změnit vzorkovací frekvenci (*Edit > Sample Rate*), nastavení hardware pomocí konfiguračního okna (*Edit > Hardware*), měřítka jednotlivých os (*Edit > Scope*), nastavení kanálu (*Edit > Channel*) a případně i měření (*Edit > Measurements*). Měřítka jednotlivých os můžeme nastavit také pomocí knoflíků v panelu *Channel Scaling*. Toto nastavení je nepříliš přesné kvůli přepočtu hodnot, avšak pokud provádíme změnu měřítka při měření, nastavení pomocí knoflíků nám pomůže nastavit nejlepší zobrazení. Nejprve před samotným měřením je tedy vhodné nastavit měřítka os. Klikneme na tlačítko *Trigger* a spustíme záznam dat. Tyto data nám slouží pouze k nastavení os. Při tomto průběhu dat pomocí knoflíků nastavíme nejlepší zobrazení. Poté klikneme na *stop* a záznam ukončíme. Abychom docílili přesného záznamu určeného počtu vzorků, musíme nastavit hodnotu *Acquire* v panelu *Trigger* na *One Shot*. Tato funkce nám zajití pouze jeden průchod. Další změnu, kterou provedeme, bude nastavení *Samples to acquire*, neboli počet pořizovaných vzorků. Přepneme *Radiobutton* (◉) z *Fill the display* do polohy *Count*, kdy bude osciloskop zaznamenávat data do získání požadovaného počtu vzorků. Tady narazíme na nevýhodu této cesty. Příkaz *Softscope* je schopen zaznamenat pouze 100 000 vzorků a to při naší frekvenci 96 000 Hz znamená, že zaznamenáme max.

okolo 1,042 s, při využití maximálního možného počtu vzorků. Pokud nám ovšem stačí záznam 1 s, je tato metoda nejlepší volbou. Nastavíme tedy hodnotu *Count* na 96 000.

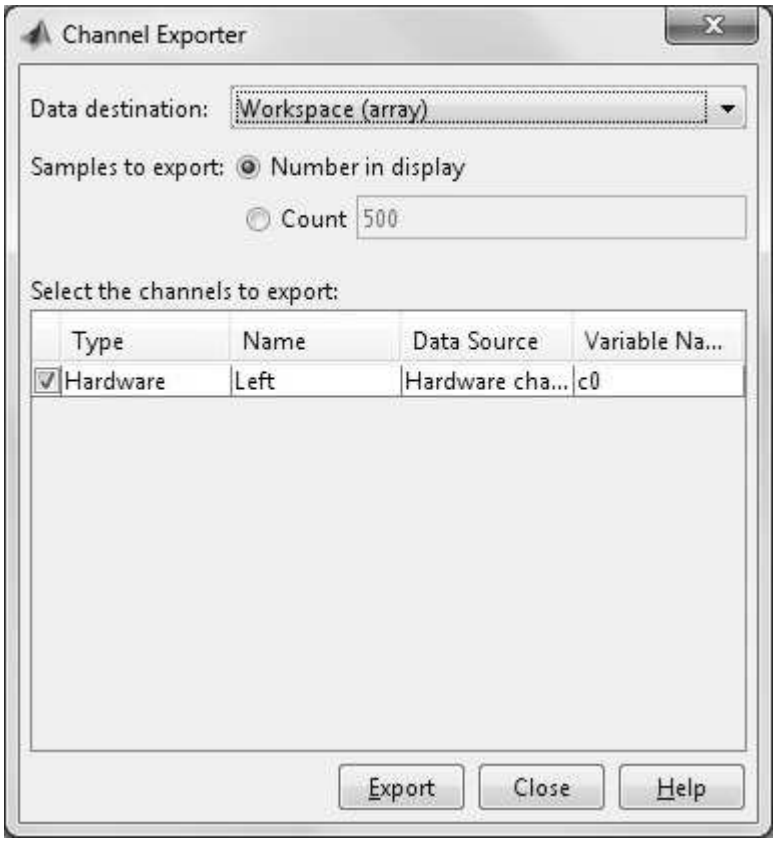


Obr. 13 Konfigurační okno příkazu *Softscope*



Obr. 14 Okno osciloskopu

Nyní po spuštění záznamu tlačítkem *Trigger* dojde k ukončení pořizování vzorků přesně po zaznamenání 96 000 vzorků. Nyní potřebujeme tyto hodnoty exportovat do samotného MATLABu, abychom mohli tyto vzorky dále zpracovávat. Tuto operaci provedeme pomocí *File* > *Export* > *Channels*. Po spuštění se nám objeví okno (Obr. 15), kde nastavíme parametry zaznamenávaných vzorků.

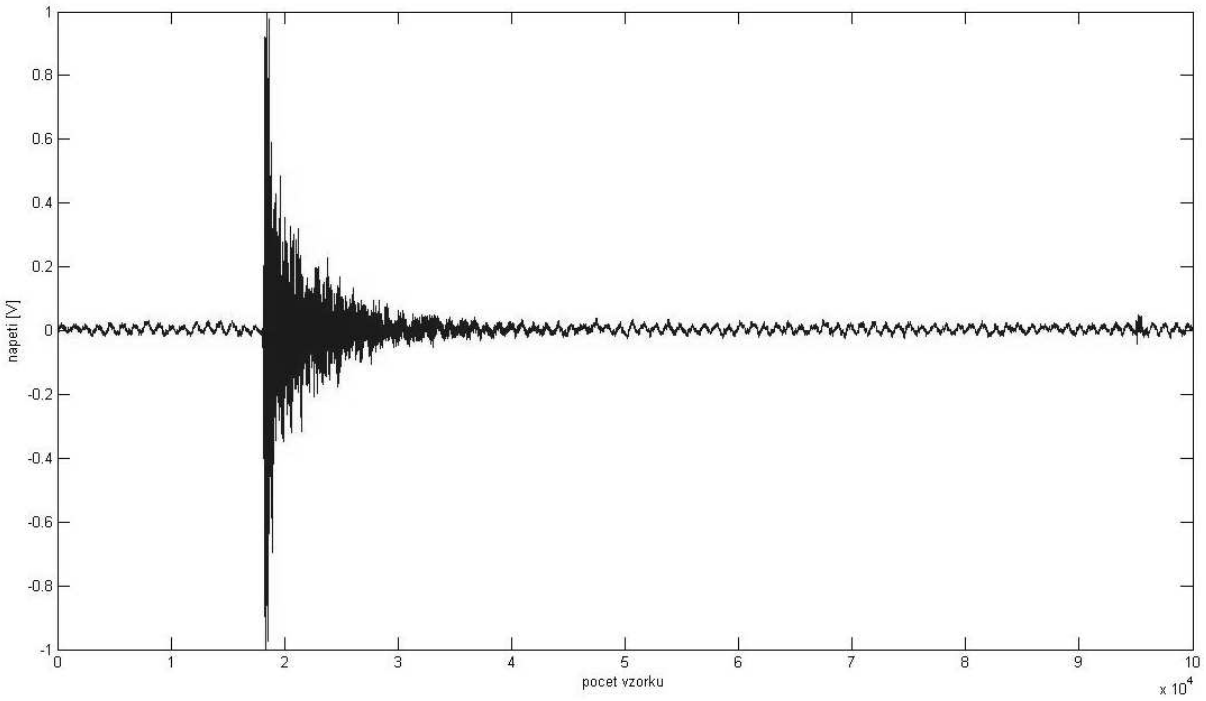


Obr. 15 Nastavení Exportu dat získaných osciloskopem

Položku *Data destination* necháme nastavenu na *Workspace (array)*, což znamená, že exportované hodnoty budou řazeny v poli. *Samples to export* opět pomocí *Radiobutton* nastavíme na *Count* na do okna vepíšeme stejnou hodnotu, kterou jsme vepisovali do *Count* v panelu *Trigger*. Tímto jsme zajistili, že exportujeme právě počet námi požadovaných vzorků (V našem případě 96 000). Nakonec nám chybí nastavit název proměnné, do které bude export proveden. Implicitně je nastaven na *c0*. Tento název najdeme ve sloupci *Variable Name*. Tam také po kliknutí na název proměnné můžeme tento název měnit. Máme-li vše nastaveno, klikneme na tlačítko *Export* a poté na tlačítko *Close*. V této chvíli můžeme okno s osciloskopem zavřít. Dalším zpracováním signálu se budeme zabývat později. Nyní si řekneme něco o druhé cestě, jak pořídit data ze zvukové karty. Touto metodou je sled příkazů nastavujících parametry pořizovaného záznamu. Nejprve musíme nastavit zvukovou kartu

jako zdroj dat. Tento příkaz je *ai=analoginput ('winsound')*. Tento příkaz vytvoří v MATLABu objekt analogového vstupu. Poté musíme vytvořit kanál vstupu pomocí příkazu *addchannel (ai,1)*, který zpřístupní kanál 1 pro objekt *ai*. Pokud za tento příkaz nenapíšeme středník, na obrazovce se nám objeví nastavení tohoto kanálu. Další nastavení, které musíme provést je nastavení vzorkovací frekvence. K tomu slouží příkaz *set(ai,'SampleRate',96000)*, tímto je nastavena pro objekt analogového vstupu *ai* na vzorkovací frekvenci 96 000 Hz. Další nastavení je počet zaznamenávaných vzorků. Použijeme opět příkaz *set*, ovšem parametry budou jiné. Příkaz bude vypadat takto: *set (ai, 'SamplesPerTrigger', 192000)*. Takto je nastaven záznam 192 000 vzorků pro objekt analogového vstupu *ai*. Nyní můžeme objekt *ai* spustit. Zadáme příkaz *start (ai)*. Nyní nám MATLAB načítá vzorky ze zvukové karty, ale nikde je nezaznamenává ani nevypisuje. Vytvoříme proto proměnnou *vzorky*, kam budeme naměřená data ukládat. Nyní máme dvě možnosti jak ukládat získávána data. Každý příkaz, který lze pro toto využít má svá specifika. Prvním příkazem je *peekdata* [1]: *„ Funkce slouží spíše k „prohlížení" dat, neblokuje běh MATLABu, vzorky z jednotky neodstraňuje a nezaručuje přenesení všech vzorků."*. Protože nechceme získaná data pouze zobrazit, ale chceme s nimi dále pracovat, použijeme funkci *getdata [1]: „Přenáší blok vzorků z jednotky zpracování dat, blokuje běh MATLABu do doby, než se podaří požadovaný počet vzorků vyčíst z jednotky (přerušit lze pomocí CTRL+C), přenesené vzorky z jednotky odstraňuje a snaží se o přenesení všech vzorků (v extrémních případech, kdy dojde ke ztrátě vzorků, se přeruší běh objektu a vyvolá funkce specifikovaná vlastností DataMissedFcn)."* Příkaz pro zápis dat do proměnné *vzorky* bude tedy vypadat takto: *vzorky=getdata(ai)*. Po každém měření je nutné, z důvodu plynulého běhu programu, daný objekt odstranit. K tomu slouží tyto dva příkazy, které se napíší za příkaz, pomocí kterého zaznamenáváme data do proměnné *vzorky*. Jsou to *delete(ai);* a *clear ai*. Výhodou tohoto postupu je, že můžeme zaznamenávat libovolný počet vzorků, který není nijak omezen (ovšem opravdu velké množství zpomaluje práci programu a v případě použití *getdata* může dojít i k nestandardnímu běhu programu). Teď si řekneme něco ke zpracování získaných dat. Někdy nám stačí pouze zobrazení zaznamenaných dat a nic více. Stačí nám tedy použít *plot*. Tato funkce slouží k vytvoření 2D grafu. Pokud bychom použili příkaz *plot3*, tak bychom vytvořili 3D graf, pokud by to bylo potřeba. Tento příkaz vytváří automaticky nové okno nazvané *Figure1, Figure 2, Figure3,…* podle toho, kolikátý graf již vytváříme. Příklad, jak mohou tato data vypadat nám ukazuje Obr. 16. Můžeme nastavovat názvy a rozměry os a mnoho dalšího. Tyto operace s funkcí *plot* nalezneme v literatuře[5]. Nás však bude zajímat jak signál vypadá ne v časové oblasti, ale v oblasti frekvenční. Fourierova transformace [13] je tzv. harmonická analýza a vyjadřuje

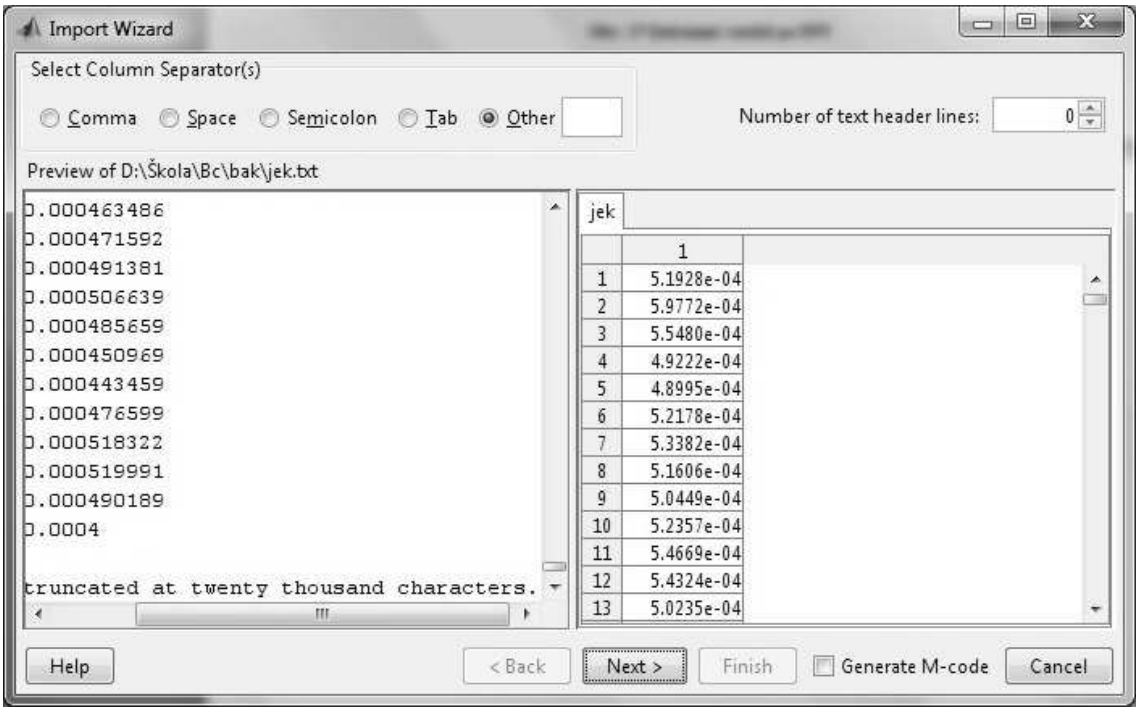
časově závislý signál pomocí harmonických signálů. Tato transformace se dá použít jak pro signály spojité, tak i pro nespojité (diskrétní). K převodu nám bude sloužit Diskrétní Fourierova transformace (DFT), která je založena na Fourierově transformaci. Diskrétní Fourierova transformace je numerická metoda, která zpracovává naměřené vzorky a určuje signál ze spektra vzorků, nebo určuje spektrum z vzorků signálu. Pro použití DFT je v počítačové technice vhodné využít algoritmus pro její spočtení a spočtení u její inverze.



Obr. 16 Zobrazení dat v časové oblasti

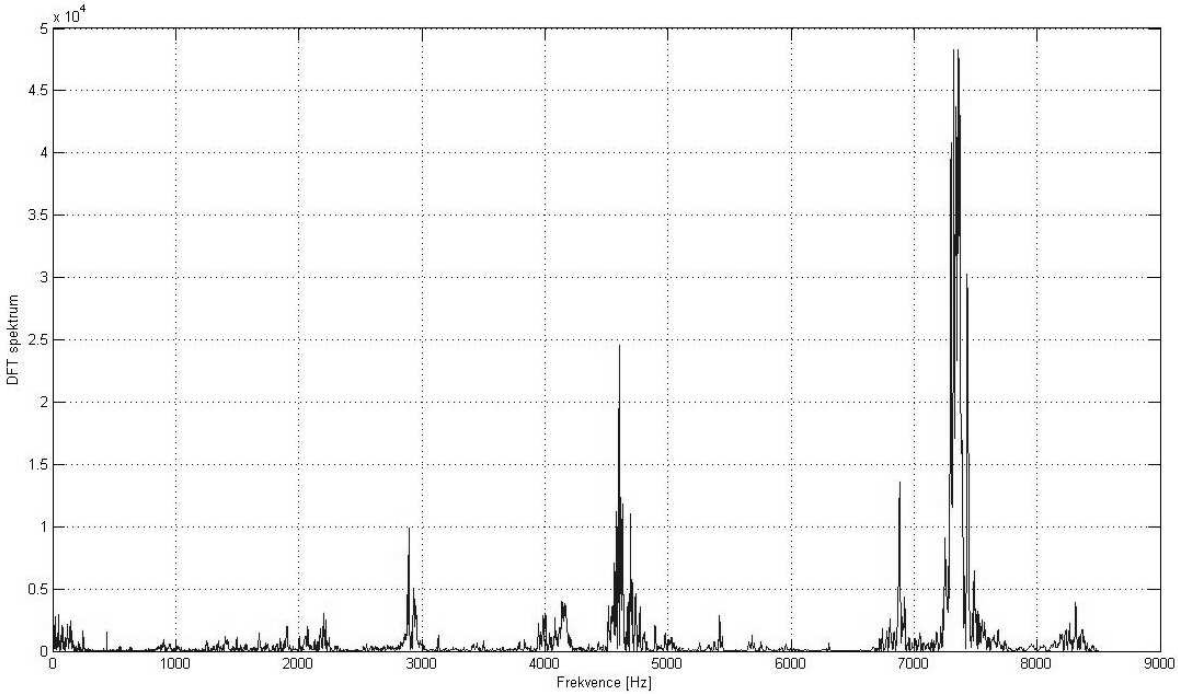
Tento algoritmus se nazývá Rychlá Fourierova transformace (Fast Fourier Transform – FFT)[9]. Když tento algoritmus použijeme v MATLABu, získáme vektor komplexních čísel. Proto musíme výsledek FFT vynásobit komplexně sdruženým vektorem. Tím docílíme „odstranění“ imaginárních částí FFT. Když toto provedeme, získáme signál ve frekvenční oblasti. Nyní si ukážeme, jak by tyto příkazy vypadaly pro náš konkrétní případ. V první řadě, si ukážeme, jak načíst vzorky ze souboru, který jsme vytvořili pomocí LabVIEW. Při spuštění MATLABu, napravo od *Command Window* nalezneme *Workspace*. V tomto okně je ikona *Import data* ( ). Když klikneme na tuto ikonu, otevře se nám okno, kde můžeme vybrat námi vytvořený soubor. Protože jsme jej uložili ve formátu *.txt, tak je viditelný pro MATLAB, který má tuto koncovku uloženu v možnosti *Recognized Data Files* (Uznávané datové soubory). Po zvolení souboru, který má být importován, se nám objeví okno *Import wizard* (viz. Obr. 17), kde můžeme nastavit, jak bude soubor importován. V našem případě

nic měnit nemusíme. Data jsou uložena ve sloupci a tak budou i importována. Poté klikneme na tlačítko *Next*. Toto okno můžeme opět ponechat beze změn, ale pokud chceme, můžeme změnit název proměnné, do které budou importovaná data uložena. Předdefinovaný název proměnné je název souboru. Název změníme tak, že dvakrát klikneme na název proměnné ve sloupci *Name* a napíšeme



Obr. 17 Import Wizard okno 1

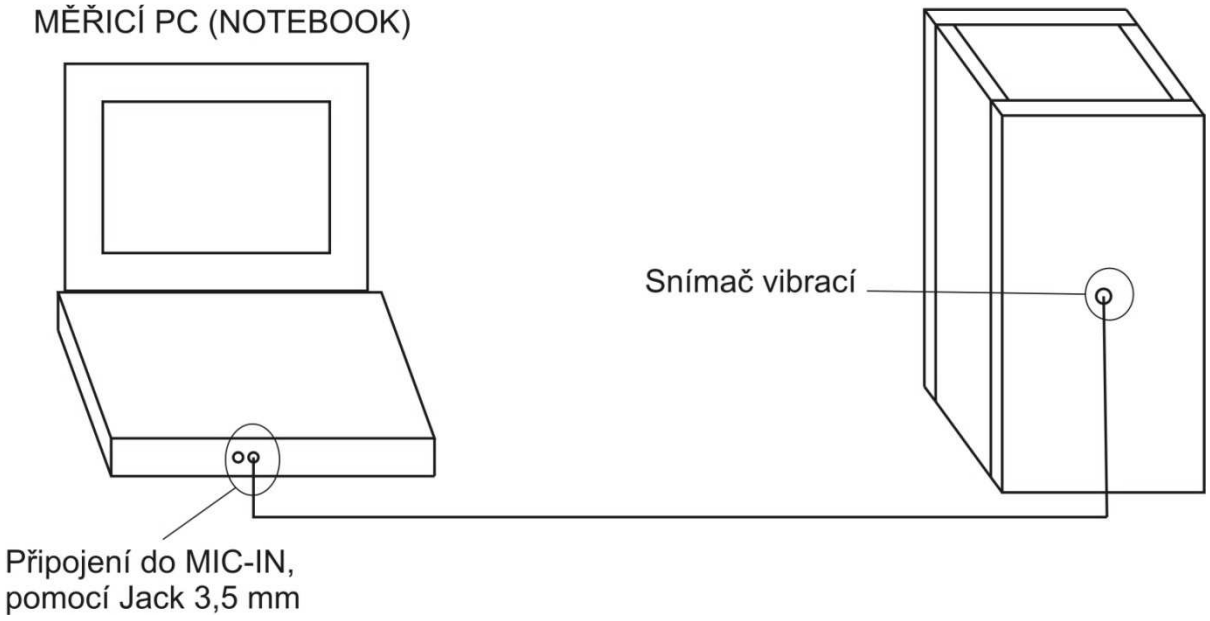
vlastní název. Poté klikneme na *Finish* a naše data jsou importována. Nyní na tyto data FFT. Příkaz je velmi jednoduchý [10] a vypadá takto: *>>x=fft(data); x* zde reprezentuje proměnnou, do které bude FFT uložena. *data* zde reprezentuje proměnnou, na kterou je FFT uplatněna. V našem případě to může být proměnná s daty, která jsme naimportovali (název souboru nebo vlastní název), nebo data získaná pomocí funkce *Softscope* (*c0* nebo vlastní název), či data získaná vytvořením objektu analogového vstupu (*vzorky* nebo vlastní název). Nyní musíme „odstranit“ imaginární části transformace. Pro vynásobení komplexně sdruženým vektorem existuje jednoduchý příkaz, který bude vypadat takto: *>> y=x.*conj(x);*. Do proměnné *y* je uložena FFT bez imaginárních částí, výraz .* slouží pro násobení prvek po prvku, *conj(x)* je komplexně sdružený vektor, kterým násobíme FFT. Nyní se nám již podařilo převést signál do frekvenční oblasti, kde může vypadat například jako na Obr. 18.



Obr. 18 Signál převedený pomocí FFT do frekvenční oblasti

3.3 Měření vibrací

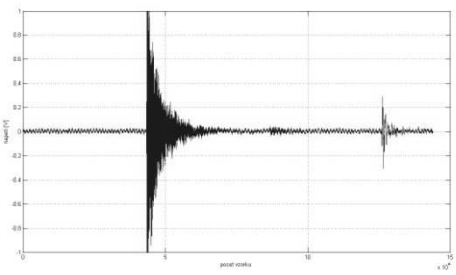
V laboratoři bylo provedeno praktické měření s využitím uvedených metod. Měření bude probíhat podle schéma zapojení na Obr. 19



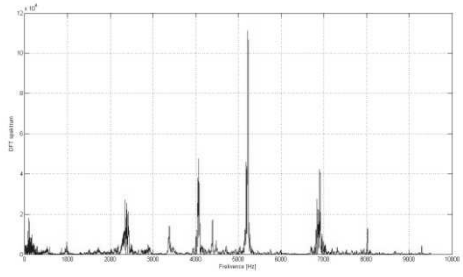
Obr. 19 Schéma připojení PC k soustavě

Soustava, na které je připojen snímač vibrací (akcelerometr), je model krystalizátoru a vibrace budeme generovat buď ručně, úderem kovové „paličky“, nebo automaticky. Nejprve provedeme záznam úderu, který provedeme ručně a to v blízkosti snímače (akcelerometru).

Data budou zaznamenána pomocí virtuálního přístroje vytvořeného v LabVIEW, který nalezneme v Příloze I. V tomto příkladu jsem použil následující nastavení. Sample rate = 96 000 Hz, Number of Channels = 1, Bits Per sample = 24, Počet vzorků = 144 000, ID zařízení = 0. Po spuštění přístroje přesunutím páčky do polohy on jsem udeřil „paličkou“ do soustavy. Poté byla zaznamenaná data uložena ve formátu *.txt (Příloha II) a virtuální přístroj vypnut. Další měření jsem provedl v programu MATLAB. Nejprve jsem rozhodl využít záznam dat pomocí příkazu *Softscope*. Z toho vyplývá, že mohu zaznamenat pouze 100 000 vzorků, a proto bude možné provést záznam pouze ručních úderů. Tento virtuální osciloskop jsem nastavil následovně. Adaptor = winsound, ID = 0, sample rate = 96 000 Hz, Input type = AC Coupled, Acquire = OneShot, Samples to acquire – count = 100 000. První měření bylo provedeno opět ručně a úder byl veden v blízkosti snímače. Toto měření bylo zaznamenáno do proměnné c0. Druhý úder byl veden do boční strany, tedy dále od snímače a byl zaznamenán do proměnné c1. V dalším měření budou údery prováděny automaticky a z tohoto důvodu nemůžeme použít virtuální osciloskop a z tohoto důvodu byl vytvořen objekt analogového vstupu, kde byly nastaveny následující hodnoty. Jeden kanál, vzorkovací frekvence 96 000 Hz, počet vzorků = 2 112 000. Tyto vzorky byly uloženy do proměnné vzorky. Údery byly situovány na stranu protější straně se snímačem. Veškerá data naleznete v Příloze III. Současně s posledním měřením, bylo provedeno kontrolní měření na specializované kartě, jehož data nalezneme v Příloze IV. Po záznamu veškerých dat jsem importoval data získaná pomocí virtuálního přístroje LabVIEW a poté jsem na jednotlivých proměnných provedl rychlou Fourierovu transformaci a získal jsem tak frekvenční spektrum úderů. V časových oblastech lze dobře pozorovat útlum vibrací po jednotlivých úderech. Na Obr. 20a je zobrazen časový průběh signálu zaznamenaného pomocí virtuálního přístroje vytvořeného v LabVIEW a v příloze V je podrobnější zobrazení. Na Obr. 20b nalezneme tytéž data již převedeny do frekvenční oblasti a v příloze VI je podrobnější zobrazení.

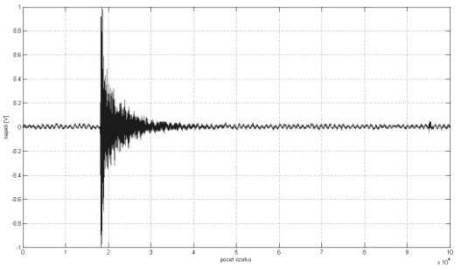


Obr. 20a Časový průběh

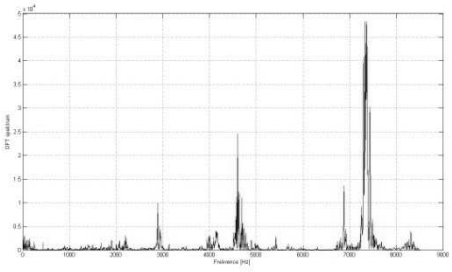


Obr. 20b Frekvenční spektrum

Na Obr. 21a můžeme sledovat časový průběh signálu uleženého v proměnné c0. Obr. 21b je frekvenční průběh tohoto signálu. V přílohách VII a VIII je podrobnější zobrazení.

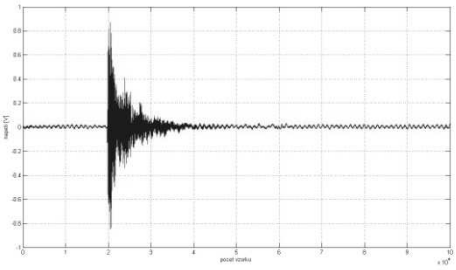


Obr. 21a Časový průběh

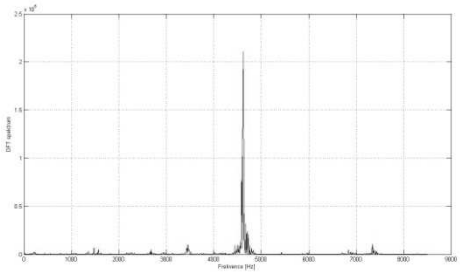


Obr. 21b Frekvenční spektrum

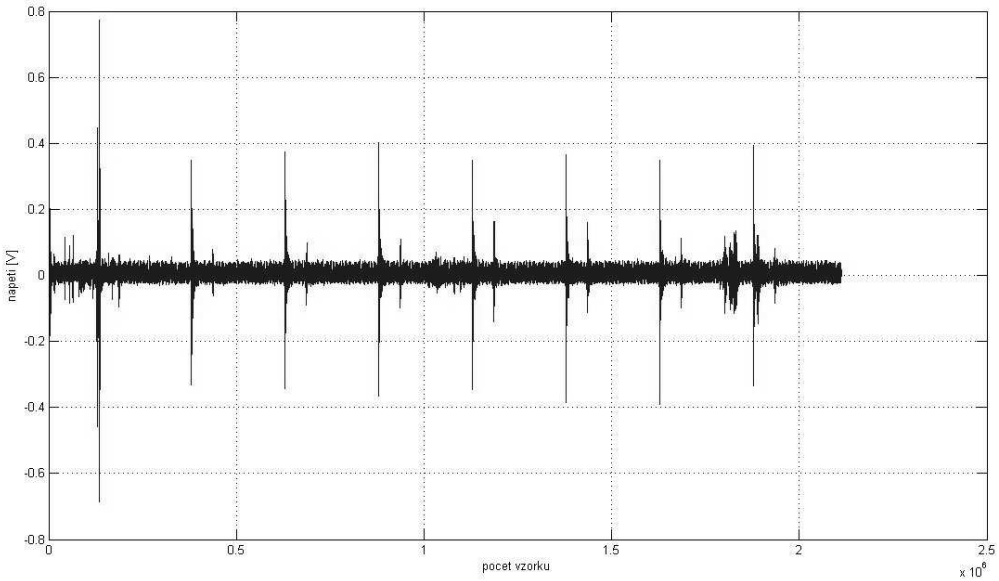
Na Obr. 22a a 22b nalezneme časové a frekvenční oblasti proměnné c1. Přílohy IX a X opět obsahují jejich podrobnější zobrazení. Na Obr. 23 je zobrazen celkový časový průběh úderů



Obr. 22a Časový průběh

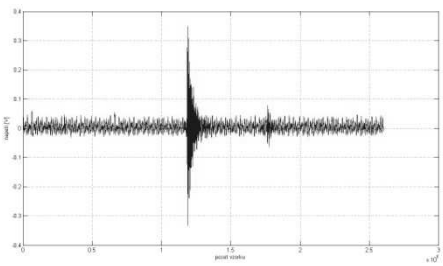


Obr. 22b Frekvenční spektrum

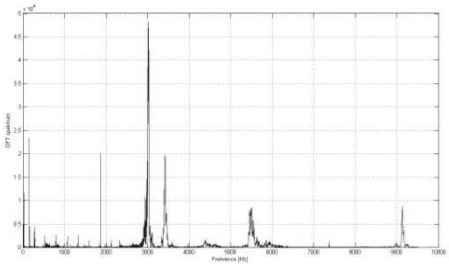


Obr. 23 Časový průběh všech vzorků uložených v proměnné vzorky

uložených v proměnné vzorky. Příloha XXVII obsahuje podrobnější zobrazení. Obr. 24a a 24b je ukázka jednoho z úderů uloženého v proměnné vzorky. Podrobnější zobrazení těchto dat a jednotlivých úderů nalezneme v přílohách XI – XVIII, které obsahují zobrazení průběhů

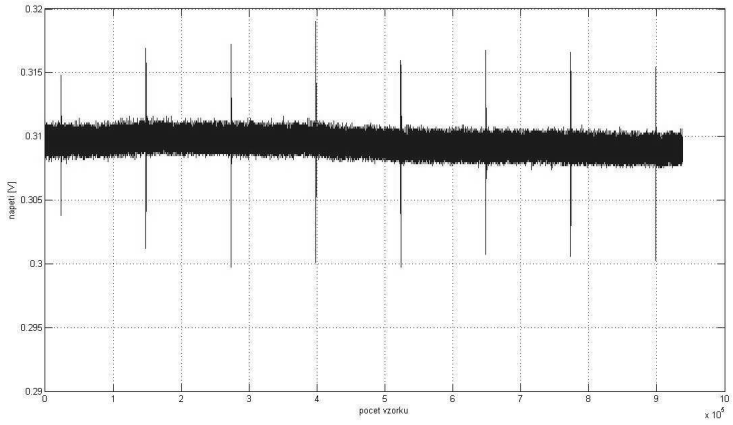


Obr. 24a Časový průběh



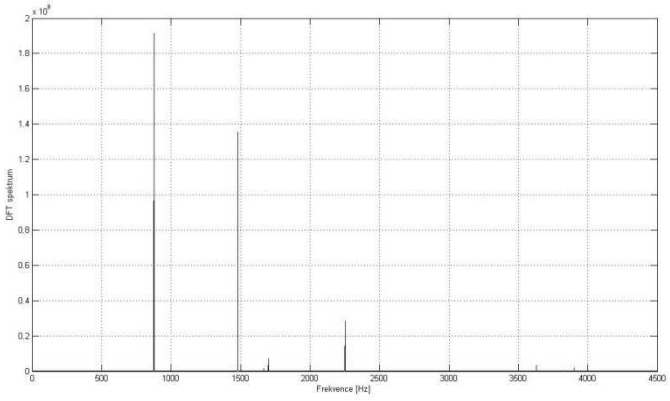
Obr. 24b Frekvenční spektrum

časových a v přílohách XIX – XXVI, které obsahují oblasti frekvenční. Na Obr. 25 nalezneme časový průběh všech úderů zaznamenaných specializovanou měřící kartou. Podrobněji v příloze XXIX.

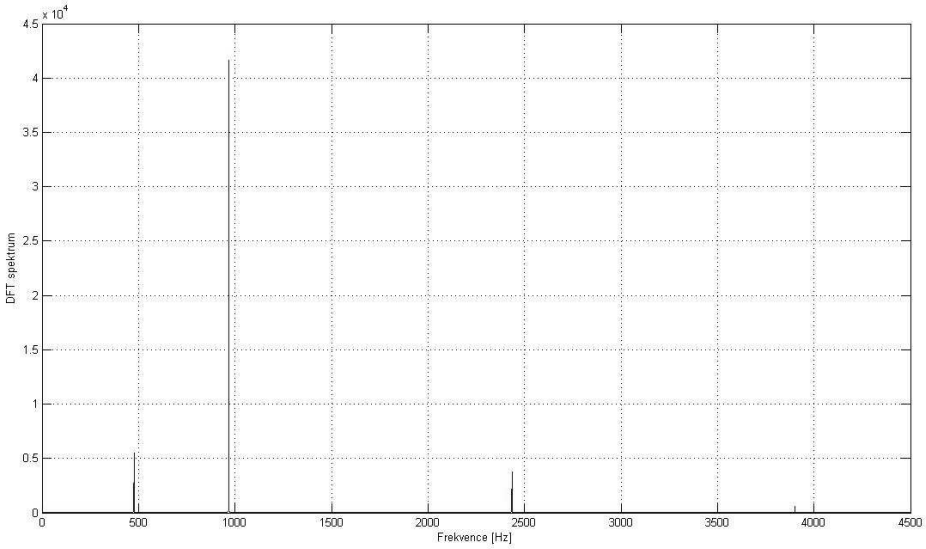


Obr. 25 Časový průběh všech úderů zaznamenaných měřicí kartou

Na Obr. 26 a 27 nalezneme srovnání frekvenčního spektra úderů zaznamenaných v proměnné vzorky a frekvenčního spektra úderů zaznamenaných pomocí specializované měřicí karty.

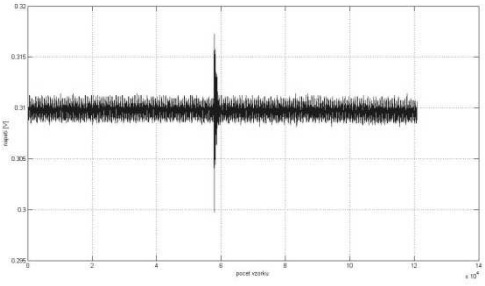


Obr. 26 Frekvenční spektrum úderů zaznamenaných v proměnné vzorky

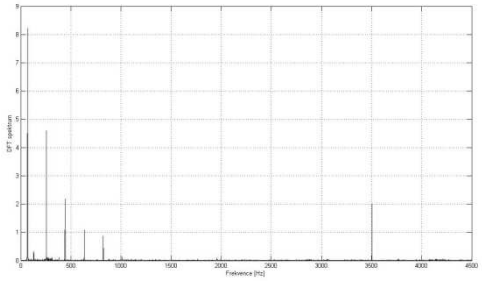


Obr. 27 Frekvenční spektrum úderů zaznamenaných měřicí kartou

Podrobnější zobrazení obsahují přílohy XXVIII a XXX. Poslední obrázky Obr. 28a a 28b zobrazují časový průběh a frekvenční spektrum jednoho úderu, který byl zaznamenán pomocí měřicí karty. Podrobnější zobrazení jednotlivých úderů obsahují přílohy XXXI až XXXVIII, kde nalezneme časové průběhy a přílohy IXL až XLVI, kde jsou uloženy grafy frekvenčních oblastí.



Obr. 28a Časový průběh



Obr. 28b Frekvenční spektrum

4. Závěr

Cílem této práce bylo zjistit možnost osobního PC jako diagnostického zařízení, bez využití speciálních měřících karet. Podle mých zkušeností získaných prací na této úloze, je využití integrovaných periferií PC k diagnostice zařízení možná a pro mnoho veličin vhodným řešením. Tento způsob měření však vyžaduje specifické zpracování signálu tak, aby byl převeden vždy na signál elektrický a byl „sveden“ do konektoru Jack 3,5 mm. Pokud bychom chtěli realizovat měření pomocí LabVIEW musíme si vždy nejprve vytvořit virtuální přístroj, který nám bude schopný vysílaná data přijmout a zpracovat. Tento přístroj může vypadat různě, protože vytvoření takovéhoto přístroje je záležitost velice individuální. Při zpracovávání dat v MATLABu je, před využitím této metody, nutno si stanovit jakou veličinu chci měřit, s jakou vzorkovací frekvencí, jakou dobu a podle toho zvolit jednu z již dříve uvedených metod. Pokud bychom chtěli porovnávat metodu měření pomocí zvukové karty a specializované měřicí karty nejlepším ukazatel by nám bylo srovnání příloh XXVIII a XXX, které obsahují frekvenční spektra těchto měření. Při srovnání těchto grafů zjistíme, že tyto metody jsou srovnatelné. Při srovnávání časových průběhů (přílohy XXVII a XXIX) zjistíme, že se tyto signály dost liší. Tato odlišnost je způsobena konstrukcí jednotlivých měřicích zařízení. Zvuková karta je specializovaná na akustické spektrum signálu a zpracovává jen signál střídavý. Měřicí karta je schopna zaznamenávat i signály stejnosměrné. Dále je patrný rozdíl velikosti amplitud. Toto je opět způsobeno rozdílnou konstrukcí. Zvuková karta má zabudovány předzesilovače signálu, a proto je amplituda větší. Také díky tomuto předzesílení můžeme pozorovat na Obr. 24a a 24b jakoby druhý malý úder. Tento je způsoben vracením se automatického úderníku, který nám generoval údery. Tento „úder“ se nám na Obr. 24b zobrazil jako frekvence cca 1500 Hz. Vzhledem k inovacím v oblasti zvukových karet, u kterých se zvyšuje nejen přesnost, ale i vzorkovací frekvence (v dnešní době jsou notebooky schopny provádět záznam s přesností 24 bit a o vzorkovací frekvenci 192 000 Hz), lze předpokládat, že tato oblast skýtá velké možnosti v oblasti diagnostiky. Oproti specializovaným kartám má tato metoda hlavní výhodu v pořizovací ceně. Je to dáno tím, že pokud chceme měřit specializovanou kartou, tak získaná data se ve většině případů také zpracovávají na PC a programech podobných LabVIEW, MATLABu, Control WEBu apod. Tyto karty představují podstatnou část finančních investic. Naproti tomu zvuková karta je v dnešní době nezbytnou součástí každého PC a v případě, že bychom si chtěli pořídit nějakou kvalitnější, jsou ceny s měřicími kartami neporovnatelné.

Seznam použité literatury

[1] SEDLÁČEK, M., ŠMÍD, R., *Matlab v měření*. Praha: ČVUT Fakulta Elektrotechnická, 2004. 204s

[2] ZAPLATÍLEK, K., DOŇAR, B., *Matlab: Začínáme se signály*. Praha: BEN – technická literatura, 2006. ISBN 80-7300-200-0.

[3] VLACH, J., HAVLÍČEK J., VLACH, M., *Začínáme s LabVIEW*. Praha: BEN – technická literatura, 2008. ISBN 978-80-7300-245-9.

[4] ZAPLATÍLEK, K., DOŇAR, B., *Matlab pro začátečníky*. 2. vydání. Praha: BEN – technická literatura, 2005. ISBN 80-7300-175-6.

[5] HERINGOVÁ, B., HORA, P., *MATLAB 4.0 – popis grafického systému, grafická nadstavba a práce se soubory Díl II. Referenční manuál*. Plzeň, červen 1994, 279 s.

[6] KUKAL, J. *Všechno má své meze* [online] Praha 2: AUTOMATIZACE s.r.o., posl. úpravy Říjen 2008 [cit. 19. 04. 2010]. Dostupné na: http://www.automatizace.cz/article.php?a=2316

[7] *Maturitní otázka z POS č.8* [online] Vyšší odborná škola, posl. úpravy neznámé [cit. 19.04.2010]. Dostupné na: http://www.ekodom.cz/ie/maturita/c-08_zobrazovaci_zvukova_soustava.pdf

[8] TESAŘ, J. *Zvukové karty* [online] Jihočeská univerzita v Českých Budějovicích – Pedagogická fakulta [cit. 25.01.2010] Dostupné na: http://www.pf.jcu.cz/stru/katedry/fyzika/prof/Tesar/diplomky/pruvodce_hw/komponenty/multimedia/zvukovka/popis.htm

[9] SMITH, S. W. *The Chapter 12: The Fast Fourier Transform* [online] The Scientist and Engineer's Guide to Digital Signal Processing, posl. úpravy 2007 [cit. 19.04.2010]. Dostupné na: http://www.dspguide.com/ch12.htm

[10] *Discrete Fourier transform - MATLAB* [online] The MathWorks, Inc. [cit. 17.02.2010]. Dostupné na: http://www.mathworks.com/access/helpdesk/help/techdoc/ref/fft.html

[11] *LabVIEW™ User Manual* [online] National Instruments™, posl. úpravy Duben 2003 [cit. 27.02.2010]. Dostupné na: http://www.ni.com/pdf/manuals/320999e.pdf

[12] *Importing Text Data Files: Importing Data (MATLAB®)* [online] The MathWorks, Inc. [cit. 01.03.2010]. Dostupné na: http://www.mathworks.com/access/helpdesk/help/techdoc/import_export/f5-35378.html

[13] MATĚJKA, J. *Úvod o multimédiích* [online] Masarykova universita – fakulta informatiky, posl. úpravy neznámé [cit. 19. 04. 2010]. Dostupné na: http://www.fi.muni.cz/~qruzicka/Matejka/Dipl-up.htm

[14] VLČEK, J. *D/A a A/D převodníky* [online] posl. úpravy neznámé [cit. 11.5.2010]. Dostupné na: http://www.tzb-info.cz/download.py?file=docu/texty/0001/000102_da_ad_prevodniky.pdf

Seznam elektronických příloh

Tyto přílohy jsou v pouze v elektronické podobě a to z důvodu jejich rozsahu a také kvůli přehlednosti, protože se zde nachází vytvořený virtuální přístroj, naměřená data a velké množství grafů, které budou přehlednější v digitální podobě. Tyto přílohy se nacházení na přiloženém CD a umístěny na adresách uvedených v tomto seznamu.

I	//LabVIEW/zaznam v2.10.vi
II	//LabVIEW/Data.txt
III	//MATLAB/vzorky.mat
IV	//MATLAB/merici_karta.mat
V	//casove_oblasti/data/datac.jpg
VI	//frekvencni_oblasti/data/dataf.jpg
VII	//casove_oblasti/c0/c0c.jpg
VIII	//frekvencni_oblasti/c0/c0f.jpg
IX	//casove_oblasti/c1/c1c.jpg
X	//frekvencni_oblasti/c1/c1f.jpg
XI	//casove_oblasti/vzorky/vzu1c.jpg
XII	//casove_oblasti/vzorky/vzu2c.jpg
XIII	//casove_oblasti/vzorky/vzu3c.jpg
XIV	//casove_oblasti/vzorky/vzu4c.jpg
XV	//casove_oblasti/vzorky/vzu5c.jpg
XVI	//casove_oblasti/vzorky/vzu6c.jpg
XVII	//casove_oblasti/vzorky/vzu7c.jpg
XVIII	//casove_oblasti/vzorky/vzu8c.jpg
XIX	//frekvencni_oblasti/vzorky/vzu1f.jpg
XX	//frekvencni_oblasti/vzorky/vzu2f.jpg
XXI	//frekvencni_oblasti/vzorky/vzu3f.jpg
XXII	//frekvencni_oblasti/vzorky/vzu4f.jpg
XXIII	//frekvencni_oblasti/vzorky/vzu5f.jpg
XXIV	//frekvencni_oblasti/vzorky/vzu6f.jpg
XXV	//frekvencni_oblasti/vzorky/vzu7f.jpg
XXVI	//frekvencni_oblasti/vzorky/vzu8f.jpg
XXVII	//casove_oblasti/vzorky/vzcc.jpg
XXVIII	//frekvencni_oblasti/vzorky/vzcf.jpg

XXIX	//casove_oblasti/merici_karta/mkcc.jpg
XXX	//frekvencni_oblasti/merici_karta/mkcf.jpg
XXXI	//casove_oblasti/merici_karta/mku1c.jpg
XXXII	//casove_oblasti/merici_karta/mku2c.jpg
XXXIII	//casove_oblasti/merici_karta/mku3c.jpg
XXXIV	//casove_oblasti/merici_karta/mku4c.jpg
XXXV	//casove_oblasti/merici_karta/mku5c.jpg
XXXVI	//casove_oblasti/merici_karta/mku6c.jpg
XXXVII	//casove_oblasti/merici_karta/mku7c.jpg
XXXVIII	//casove_oblasti/merici_karta/mku8c.jpg
IXL	//frekvencni_oblasti/merici_karta/mku1f.jpg
XL	//frekvencni_oblasti/merici_karta/mku2f.jpg
XLI	//frekvencni_oblasti/merici_karta/mku3f.jpg
XLII	//frekvencni_oblasti/merici_karta/mku4f.jpg
XLIII	//frekvencni_oblasti/merici_karta/mku5f.jpg
XLIV	//frekvencni_oblasti/merici_karta/mku6f.jpg
XLV	//frekvencni_oblasti/merici_karta/mku7f.jpg
XLVI	//frekvencni_oblasti/merici_karta/mku8f.jpg